ADAS User Callable Subroutine Library Documentation

A D Whiteford, M G O'Mullane and H P Summers

Generated on December 23, 2008

Contents

1 Introduction

This document describes the user callable ADAS subroutine libraries which exist inside various ADAS libraries. These routines may be linked in to user written FORTRAN codes.

For subroutine naming, codes series 1-8 are labelled alphabetically a-h. A subroutine associated with series 1 begins with the letter 'a', a subroutine associated with series 2 begins with letter 'b' and so on. A subroutine associated with a particular member of a code series has the code number as the second character of the subroutine name. Thus the name of a subroutine associated with the code ADAS405, such as `d5popm`, begins with the letters 'd5'. The name of a general subroutine from a code series but not restricted to a specific code has 'x' as the second character of its name. For example `bxdata` is a general series 2 subroutine. A general subroutine, not restricted to a single code series, begins with the letters 'xx' (e.g. `xxuid`). General functions are preceded by 'i4', 'r8', 'c' depending on the type of function value returned. It should be noted that a significant number of older subroutines have not yet been converted to the above naming systems and retain arbitrary, partly descriptive names. **Note that to preserve the integrity of ADAS, FORTRAN subroutine source code is not accessible to the general user except for agreed development purposes.** The headers of the FORTRAN subroutines, including call parameters, description of the routine and all variables and declaration statements are provided in this document for the various ADAS libraries. The FORTRAN scannable object libraries are located as follow:

- `/.../adas/lib/libadas1xx.a` 'ax', 'a1', 'a2', ...
- `/.../adas/lib/libadas2xx.a` 'bx', 'b1', 'b2', ...
- `/.../adas/lib/libadas3xx.a` 'cx', 'c1', 'c2', ...
- `/.../adas/lib/libadas4xx.a` 'dx', 'd1', 'd2', ...
- `/.../adas/lib/libadas5xx.a` 'ex', 'e1', 'e2', ...
- `/.../adas/lib/libadas7xx.a` 'gx', 'g1', 'g2', ...
- `/.../adas/lib/libadas8xx.a` 'hx', 'h1', 'h2', ...
- `/.../adas/lib/libadaslib.a` 'xx','i4', 'r8' and 'c'...

Note that only subroutines judged to be of use to external programs are included in these libraries, subroutines only relevant to the interactive nature of ADAS are not included. Practical information on how to include subroutines in user written code is given in section 10.

2 Subroutine library adas1xx

2.1 a1data: Subroutine a1data from library adas1xx

```
       SUBROUTINE A1DATA( DSFULL , INDXREF , TITLE , CAMETH , Z0    ,
     &                    Z      , ZEFF    , INDL  , INDU   , EI    ,
     &                    EJ     , WI      , WJ    , ACOEFF , S     ,
     &                    FIJ    , EIJ     , IXTYP , FXC2   , FXC3  ,
     &                    IXOPS  , IBPTS   , IFPTS , IDIFF  , ICT   ,
     &                    ITOUT  , XA , YA , APOMA , DIFOMA , TOA   ,
     &                    GOA    , APGOA   , EXCRA , DEXCRA , GBARFA,
     &                    ISTDIM , IREAD   , IZ    , IZ0    , GF    ,
     &                    BCVAL
     &                )
C-----------------------------------------------------------------------
C
C  ********************* FORTRAN77 SUBROUTINE A1DATA ******************
C
C  PURPOSE: TO REFRESH A DATA INDEX FROM AN ADAS101 ARCHIVE. READS
C           IN THE INDEX CODE A-ADAS, B-BURGESS AND THE THE REST OF
C           THE DATA AS APPROPRIATE.
C
C  CALLING PROGRAM:
C            ADAS101.FOR
C
C  INPUT:
C            (C*80) DSFULL   - THE USERS' CHOSEN ARCHIVE FILE NAME.
C            (I*4)  INDXREF  - THE INDEX NUMBER TO REFRESH FROM.
C            (C*40) TITLE    - THE INFORMATION LINE IN THE ARCHIVE
C                              FILE.
C            (C*4)  CAMETH   - THE TAG TO DISTINGUISH BETWEEN THE
C                              TWO TYPES OF ANALYSIS.
C                              A - ADAS, B- BURGESS
C            (R*8)  GF       - THE WEIGHTED OSCILLATOR STRENGTH
C            (R*8)  BCVAL    - THE BURGESS SCALABLE PARAMETER C.
C            (I*4)  ISTDIM = THE MAXIMUM ARRAY DIMENSION
C            (I*4)  IREAD  = THE INPUT UNIT
C
C  OUTPUTS:
C            (R*8)  Z0     = NUCLEAR CHARGE OF ION
C            (R*8)  Z      = ION CHARGE
C            (R*8)  ZEFF   = ION CHARGE + 1
C            (I*4)  INDL   = LOWER LEVEL INDEX  (USER CHOICE)
C            (I*4)  INDU   = UPPER LEVEL INDEX  (USER CHOICE)
C            (R*8)  WI     = LOWER LEVEL STATISTICAL WEIGHT
C            (R*8)  WJ     = UPPER LEVEL STATISTICAL WEIGHT
C            (R*8)  EI     = LOWER LEVEL ENERGY (IN SELECTED UNITS)
C            (R*8)  EJ     = UPPER LEVEL ENERGY
C            (R*8)  ACOEFF = TRANSITION PROBABILITY (IN ABOVE FORM,
C                            DIPOLE CASE ONLY)
C            (I*4)  IXTYP  = 1  DIPOLE TRANSITION
C                          = 2  NON-DIPOLE TRANSITION
C                          = 3  SPIN CHANGE TRANSITION
C                          = 4  OTHER
C            (I*4)  IBPTS  = 0  BAD POINT OPTION OFF
C                          = 1  BAD POINT OPTION ON
C            (I*4)  IFPTS  = 1  SELECT ONE POINT OPTIMISING
C                          = 2  SELECT TWO POINT OPTIMISING
C            (I*4)  IXOPS  = 0  OPTIMISING OFF
C                          = 1  OPTIMISING ON  (IF ALLOWED)
C            (I*4)  IDIFF  = 0  RATIO FITTING FOR DIPOLE X-SECT(ONLY
```

```
C                              WITH OPTIMISING)
C                           = 1  DIFFERENCE FITTING FOR DIPOLE X-SECT
C            (R*8)  S      = LINE STRENGTH
C            (R*8)  FIJ    = OSCILLATOR STRENGTH
C            (R*8)  EIJ    = TRANSITION ENERGY
C            (R*8)  FXC2   = SPLINING VARIABLE
C            (R*8)  FXC3   = SPLINING VARIABLE
C            (I*4)  ICT    = NUMBER OF X-SECTIONS
C            (I*4)  ITOUT  = NUMBER OF TEMPERATURES
C            (R*8)  XA     = ENERGY (PARAMETER X)
C            (R*8)  YA     = OMEGA (COLLISION STRENGTH)
C            (R*8)  APOMA  = APPROXIMATE OMEGA
C            (R*8)  DIFOMA = DIFFERENCE BETWEEN YA & APOMA
C            (R*8)  TOA    = TEMPERATURE SET
C            (R*8)  GOA    = GAMMA (EFFECTIVE COLLISION STRENGTHS)
C            (R*8)  APGOA  = APPROXIMATE GAMMA
C            (R*8)  EXCRA  = EXCITATION RATE COEFFICIENT
C            (R*8)  DEXCRA = DEEXCITATION RATE COEFFICIENT
C            (R*8)  GBARFA = G BAR FUNCTION
C            (I*4)  ISTDIM = THE MAXIMUM ARRAY DIMENSION
C            (I*4)  IREAD  = THE INPUT UNIT
C
C  ROUTINES: NONE
C
C  AUTHOR:   DAVID H.BROOKS (UNIV.OF STRATHCLYDE) EXT.4213/4205
C
C  VERSION 1.1                              DATE:     26/05/95
C  MODIFIED: DAVID H. BROOKS
C - FIRST RELEASE
C  VERSION 1.2                              DATE:     03/10/96
C  MODIFIED: WILLIAM OSBORN
C        - ADDED TRAP FOR WHEN THE REQUESTED ARCHIVE NUMBER IS NOT IN
C          THE FILE
C
C  DATE: 07/05/99 VERSION 1.3
C  MODIFIED: HUGH SUMMERS
C - CORRECTED CONFUSION ABOUT NCHAR AND NELEC
C
C-----------------------------------------------------------------------
      CHARACTER*4         CAMETH
      CHARACTER*80        DSFULL
      CHARACTER*40        TITLE
      INTEGER             IBPTS,       ICT,         IDIFF,       IFPTS
      INTEGER             INDL,        INDU,        INDXREF,     IREAD
      INTEGER             ISTDIM,      ITOUT,       IXOPS,       IXTYP
      INTEGER             IZ,          IZ0
      REAL*8              ACOEFF,      APGOA(ISTDIM)
      REAL*8              APOMA(ISTDIM),            BCVAL
      REAL*8              DEXCRA(ISTDIM),           DIFOMA(ISTDIM)
      REAL*8              EI,          EIJ,         EJ
      REAL*8              EXCRA(ISTDIM),            FIJ,         FXC2
      REAL*8              FXC3,        GBARFA(ISTDIM),           GF
      REAL*8              GOA(ISTDIM), S,           TOA(ISTDIM), WI
      REAL*8              WJ,          XA(ISTDIM),  YA(ISTDIM),  Z
      REAL*8              Z0,          ZEFF
```

2.2 a2data: Subroutine a2data from library adas1xx

```
       SUBROUTINE A2DATA( DSFULL , INDXREF , TITLE , CAMETH , Z0    ,
     &                    Z      , ZEFF    , INDL  , INDU   , EI    ,
     &                    EJ     , WI      , WJ    , ACOEFF , S     ,
     &                    FIJ    , EIJ     , IXTYP , FXC2   , FXC3  ,
     &                    IXOPS  , IBPTS   , IFPTS , IDIFF  , ICT   ,
     &                    ITOUT  , XA , YA , APOMA , DIFOMA , TOA   ,
     &                    GOA    , APGOA   , EXCRA , DEXCRA , GBARFA,
     &                    ISTDIM , IREAD   , IZ    , IZ0    , GF    ,
     &                    BCVAL
     &                 )
C-----------------------------------------------------------------------
C
C  ********************* FORTRAN77 SUBROUTINE A2DATA ******************
C
C  PURPOSE: TO REFRESH A DATA INDEX FROM AN ADAS102 ARCHIVE. READS
C           IN THE INDEX CODE A-ADAS, B-BURGESS AND THE THE REST OF
C           THE DATA AS APPROPRIATE.
C
C  CALLING PROGRAM:
C            ADAS102.FOR
C
C  INPUT:
C            (C*80) DSFULL   - THE USERS' CHOSEN ARCHIVE FILE NAME.
C            (I*4)  INDXREF  - THE INDEX NUMBER TO REFRESH FROM.
C            (C*40) TITLE    - THE INFORMATION LINE IN THE ARCHIVE
C                              FILE.
C            (C*4)  CAMETH   - THE TAG TO DISTINGUISH BETWEEN THE
C                              TWO TYPES OF ANALYSIS.
C                              A - ADAS, B- BURGESS
C            (R*8)  GF       - THE WEIGHTED OSCILLATOR STRENGTH
C            (R*8)  BCVAL    - THE BURGESS SCALABLE PARAMETER C.
C            (I*4)  ISTDIM = THE MAXIMUM ARRAY DIMENSION
C            (I*4)  IREAD  = THE INPUT UNIT
C
C  OUTPUTS:
C            (R*8)  Z0     = NUCLEAR CHARGE OF ION
C            (R*8)  Z      = ION CHARGE
C            (R*8)  ZEFF   = ION CHARGE + 1
C            (I*4)  INDL   = LOWER LEVEL INDEX  (USER CHOICE)
C            (I*4)  INDU   = UPPER LEVEL INDEX  (USER CHOICE)
C            (R*8)  WI     = LOWER LEVEL STATISTICAL WEIGHT
C            (R*8)  WJ     = UPPER LEVEL STATISTICAL WEIGHT
C            (R*8)  EI     = LOWER LEVEL ENERGY (IN SELECTED UNITS)
C            (R*8)  EJ     = UPPER LEVEL ENERGY
C            (R*8)  ACOEFF = TRANSITION PROBABILITY (IN ABOVE FORM,
C                            DIPOLE CASE ONLY)
C            (I*4)  IXTYP  = 1  DIPOLE TRANSITION
C                          = 2  NON-DIPOLE TRANSITION
C                          = 3  SPIN CHANGE TRANSITION
C                          = 4  OTHER
C            (I*4)  IBPTS  = 0  BAD POINT OPTION OFF
C                          = 1  BAD POINT OPTION ON
C            (I*4)  IFPTS  = 1  SELECT ONE POINT OPTIMISING
C                          = 2  SELECT TWO POINT OPTIMISING
C            (I*4)  IXOPS  = 0  OPTIMISING OFF
C                          = 1  OPTIMISING ON  (IF ALLOWED)
C            (I*4)  IDIFF  = 0  RATIO FITTING FOR DIPOLE X-SECT(ONLY
C                               WITH OPTIMISING)
C                          = 1  DIFFERENCE FITTING FOR DIPOLE X-SECT
```

```
C             (R*8)  S      = LINE STRENGTH
C             (R*8)  FIJ    = OSCILLATOR STRENGTH
C             (R*8)  EIJ    = TRANSITION ENERGY
C             (R*8)  FXC2   = SPLINING VARIABLE
C             (R*8)  FXC3   = SPLINING VARIABLE
C             (I*4)  ICT    = NUMBER OF X-SECTIONS
C             (I*4)  ITOUT  = NUMBER OF TEMPERATURES
C             (R*8)  XA     = ENERGY (PARAMETER X)
C             (R*8)  YA     = OMEGA (COLLISION STRENGTH)
C             (R*8)  APOMA  = APPROXIMATE OMEGA
C             (R*8)  DIFOMA = DIFFERENCE BETWEEN YA & APOMA
C             (R*8)  TOA    = TEMPERATURE SET
C             (R*8)  GOA    = GAMMA (EFFECTIVE COLLISION STRENGTHS)
C             (R*8)  APGOA  = APPROXIMATE GAMMA
C             (R*8)  EXCRA  = EXCITATION RATE COEFFICIENT
C             (R*8)  DEXCRA = DEEXCITATION RATE COEFFICIENT
C             (R*8)  GBARFA = G BAR FUNCTION
C             (I*4)  ISTDIM = THE MAXIMUM ARRAY DIMENSION
C             (I*4)  IREAD  = THE INPUT UNIT
C
C  ROUTINES: NONE
C
C  AUTHOR:   HUGH P. SUMMERS, UNIVERSITY OF STRATHCLYDE
C            TEL 0141-553-4196
C
C  VERSION 1.1                              DATE:     15/11/96
C - FIRST RELEASE
C
C  VERSION 1.2    DATE:     18/05/99
C  MODIFIED: HUGH SUMMERS
C - CORRECTED CONFUSION ABOUT NCHAR AND NELEC
C
C-----------------------------------------------------------------------
      CHARACTER*4         CAMETH
      CHARACTER*80        DSFULL
      CHARACTER*40        TITLE
      INTEGER             IBPTS,       ICT,         IDIFF,       IFPTS
      INTEGER             INDL,        INDU,        INDXREF,     IREAD
      INTEGER             ISTDIM,      ITOUT,       IXOPS,       IXTYP
      INTEGER             IZ,          IZ0
      REAL*8              ACOEFF,      APGOA(ISTDIM)
      REAL*8              APOMA(ISTDIM),            BCVAL
      REAL*8              DEXCRA(ISTDIM),           DIFOMA(ISTDIM)
      REAL*8              EI,          EIJ,         EJ
      REAL*8              EXCRA(ISTDIM),            FIJ,         FXC2
      REAL*8              FXC3,        GBARFA(ISTDIM),           GF
      REAL*8              GOA(ISTDIM), S,           TOA(ISTDIM), WI
      REAL*8              WJ,          XA(ISTDIM),  YA(ISTDIM),  Z
      REAL*8              Z0,          ZEFF
```

2.3 a3data: Subroutine a3data from library adas1xx

```
      SUBROUTINE A3DATA( DSFULL , INDXREF , TITLE , CAMETH , Z0    , Z ,
     &                   Z1     , N0, V0, PHFRAC,
     &                   IXOPT, IBPOPT, IFSEL, IBPTS,
     &                   EDISPO, SCALEO,
     &                   NIA, LIA, NJA, LJA, NCUTA, WIA, WJA, EIJA,FIJA,
     &                   CORFIA,
     &                   XA, YA, XOA,
     &                   NGROUP, IGROUP, ICT, ICOUT,
     &                   IREAD
     &     )
C-----------------------------------------------------------------------
C
C  ******************** FORTRAN77 SUBROUTINE A3DATA ******************
C
C  PURPOSE: TO REFRESH A DATA INDEX FROM AN ADAS105 ARCHIVE. READS
C           IN THE INDEX CODE A-ADAS, B-BURGESS AND THE THE REST OF
C           THE DATA AS APPROPRIATE.
C
C  CALLING PROGRAM:
C            ADAS105.FOR
C
C  INPUT:
C            (C*80) DSFULL   - THE USERS' CHOSEN ARCHIVE FILE NAME.
C            (I*4)  INDXREF  - THE INDEX NUMBER TO REFRESH FROM.
C            (C*4)  CAMETH   - THE TAG TO DISTINGUISH BETWEEN THE
C                              TWO TYPES OF ANALYSIS.
C                              A - ADAS, B- BURGESS
C            (I*4)  IREAD  = THE INPUT UNIT
C
C  OUTPUTS:
C            (C*40) TITLE    - THE INFORMATION LINE IN THE ARCHIVE
C                              FILE.
C     (R*8)  Z0    NUCLEAR CHARGE
C     (R*8)  Z1    RECOMBINING ION CHARGE
C     (R*8)  Z     RECOMBINED ION CHARGE
C     (I*4)  N0     = LOWEST ACCESSIBLE PRINCIPLE QUANTUM NO.
C     (R*8)  V0     = LOWEST ACCESSIBLE EFF. PRINCIPLE QUANTUM NO.
C     (R*8)  PHFRAC = LOWEST ACCESSIBLE PHASE OCCUPATION FACTOR
C     (C*40) TITLE  = TITLE FOR RUN
C     (I*4)  ICT    = NUMBER OF TEMP./RATE PAIRS
C     (I*4)  ICOUT  = NUMBER OF OUTPUT TEMPS
C     (I*4)  IBPOPT = OPTIMISE BURGESS FORMULA FIT? 1=YES 0=NO
C     (I*4)  IXOPT  = OPTIMISE BURGESS PROGRAM FIT? 1=YES 0=NO
C     (I*4)  IFSEL  = 0=FIT TO FORMULA, 1=FIT TO INPUT DATA
C     (I*4)  IBPTS  = BAD POINT OPTION 1=ON 0=OFF
C     (I*4)  NIA(,) = NI VALUES FOR BOTH GROUPS
C     (I*4)  LIA(,) = LI VALUES FOR BOTH GROUPS
C     (I*4)  NJA(,) = NJ VALUES FOR BOTH GROUPS
C     (I*4)  LJA(,) = LJ VALUES FOR BOTH GROUPS
C     (I*4)  NCUTA(,)= NCUT VALUES FOR BOTH GROUPS
C     (R*8)  WIA(,) = WI VALUES FOR BOTH GROUPS
C     (R*8)  EIJA(,)= EIJ VALUES FOR BOTH GROUPS
C     (R*8)  FIJA(,)= FIJ VALUES FOR BOTH GROUPS
C     (R*8)  CORFIA(,)=INITIAL CORFAC VALUES FOR BOTH GROUPS
C     (R*8)  XA()   = INPUT TEMPERATURE FROM ARCHIVE
C     (R*8)  YA()   = RATE FROM ARCHIVE
C     (R*8)  XOA()  = OUTPUT TEMPERATURE FROM ARCHIVE
C     (I*4)  NGROUP = NUMBER OF CORE TRANSITION GROUPS
C     (I*4)  IGROUP()=NUMBER OF ENTRIES FOR EACH GROUP
```

```
C
C  ROUTINES: NONE
C
C  AUTHOR:   WILLIAM OSBORN, TESSELLA SUPPORT SERVICES PLC., 6TH NOV 1996
C
C  VERSION 1.1                                          DATE: 06-11-96
C  MODIFIED: WILLIAM OSBORN
C - FIRST RELEASE
C
C-----------------------------------------------------------------------
      CHARACTER*4         CAMETH
      CHARACTER*80        DSFULL
      CHARACTER*40        TITLE
      INTEGER             IBPOPT,      IBPTS,       ICOUT,       ICT
      INTEGER             IFSEL,       IGROUP(2),   INDXREF,     IREAD
      INTEGER             IXOPT,       LIA(2,6),    LJA(2,6),    N0
      INTEGER             NCUTA(2,6),  NGROUP,      NIA(2,6)
      INTEGER             NJA(2,6)
      REAL*8              CORFIA(2,6), EDISPO(2),   EIJA(2,6)
      REAL*8              FIJA(2,6),   PHFRAC,      SCALEO(2),   V0
      REAL*8              WIA(2,6),    WJA(2,6),    XA(10),      XOA(10)
      REAL*8              YA(10),      Z,           Z0,          Z1
```

2.4 a5data: Subroutine a5data from library adas1xx

```
      SUBROUTINE A5DATA( DSFULL , INDXREF , TITLE , CAMETH , Z0    , Z ,
     &                   Z1     , NIGRP   , EMIN  , CIA    , NSHELA,
     &                   EIONA  , IZETAA  , NRGRP , CRA    , NRESOA,
     &                   ENERA  , WGHTA   , ICT   , XA     , YA    ,APA,
     &                   ITOUT   , TOA   , YOA    , YOAP  ,
     &                   ISTDIM , IREAD   , NA    , LA
     &     )
C-----------------------------------------------------------------------
C
C  ******************** FORTRAN77 SUBROUTINE A5DATA ******************
C
C  PURPOSE: TO REFRESH A DATA INDEX FROM AN ADAS105 ARCHIVE. READS
C           IN THE INDEX CODE A-ADAS, B-BURGESS AND THE THE REST OF
C           THE DATA AS APPROPRIATE.
C
C  CALLING PROGRAM:
C            ADAS105.FOR
C
C  INPUT:
C            (C*80) DSFULL   - THE USERS' CHOSEN ARCHIVE FILE NAME.
C            (I*4)  INDXREF  - THE INDEX NUMBER TO REFRESH FROM.
C            (C*4)  CAMETH   - THE TAG TO DISTINGUISH BETWEEN THE
C                              TWO TYPES OF ANALYSIS.
C                              A - ADAS, B- BURGESS
C            (I*4)  ISTDIM = ARRAY DIMENSIONS : MAX. NO OF VALUES THAT
C                            CAN BE READ IN
C            (I*4)  IREAD  = THE INPUT UNIT
C
C  OUTPUTS:
C            (C*40) TITLE    - THE INFORMATION LINE IN THE ARCHIVE
C                              FILE.
C            (R*8)  Z0     = NUCLEAR CHARGE OF ION
C            (R*8)  Z      = INITIAL ION CHARGE
C            (R*8)  Z1     = FINAL ION CHARGE
C            (I*4)  NIGRP  = NO. OF SHELL GROUPS
C            (R*8)  EMIN   = MINIMUM ENERGY (?)
C            (R*8)  CIA()  = SCALING PARAMETERS FOR SHELL GROUPS
C            (I*4)  NSHELA()=NO. OF ENTRIES FOR EACH SHELL GROUP
C            (I*4)  NA(,)  = SHELL GROUP DATA : N
C            (I*4)  LA(,)  = SHELL GROUP DATA : L
C            (R*8)  EIONA(,)=SHELL GROUP DATA : EION(RYD)
C            (I*4)  IZETAA(,)=SHELL GROUP DATA : IZETA
C            (I*4)  NRGRP  = NO. OF RESONANCE GROUPS
C            (R*8)  CRA()  = SCALING PARAMETERS FOR RESONANCE GROUPS
C            (I*4)  NRESOA()=NO. OF ENTRIES FOR EACH RESONANCE GROUP
C            (R*8)  ENERA(,)=RESONANCE GROUP DATA : ENERGY(RYD)
C            (R*8)  WGHTA(,)=RESONANCE GROUP DATA : WEIGHT
C            (R*8)  ICT    = NO. OF ENERGY / X-SECTION PAIRS
C            (R*8)  XA()   = X, THRESHOLD PARAMETER RELATIVE TO FIRST IONIS. POT.
C            (R*8)  YA()   = Q/(1-1/X), Q=CROSS-SECTION ?
C            (R*8)  APA()  = QEM/(1-1/X), QEM=APPROX. X-SECTION )
C            (R*4)  YPA()  = Q/QEM
C            (I*4)  ITOUT  = NO. OF TEMPS.
C            (R*8)  TOA()  = TEMP (KELVIN)
C            (R*8)  YOA()  = S, MAXWELL AVERAGED IONISATION RATE COEFF.(CM^3 S^-1)
C            (R*8)  YOAP() = SEM, APPROXIMATE RATE COEFF.
C
C  ROUTINES: NONE
C
```

```
C  AUTHOR:   WILLIAM OSBORN, TESSELLA SUPPORT SERVICES PLC., 28TH AUG 1996
C
C  VERSION 1.1                                              DATE: 28-08-96
C  MODIFIED: WILLIAM OSBORN
C - FIRST RELEASE
C
C  VERSION 1.2                                              DATE: 08-10-96
C  MODIFIED: WILLIAM OSBORN
C - REMOVED FILE READ FOR NRGRP=0 CASE
C               - ADDED READ OF NA AND LA FROM ARCHIVE
C-----------------------------------------------------------------------
      CHARACTER*4         CAMETH
      CHARACTER*80        DSFULL
      CHARACTER*40        TITLE
      INTEGER             ICT,          INDXREF,     IREAD,       ISTDIM
      INTEGER             ITOUT,        IZETAA(6,2), LA(6,2),     NA(6,2)
      INTEGER             NIGRP,        NRESOA(2),   NRGRP
      INTEGER             NSHELA(2)
      REAL*8              APA(ISTDIM), CIA(2),      CRA(2)
      REAL*8              EIONA(6,2),  EMIN,        ENERA(6,2)
      REAL*8              TOA(ISTDIM), WGHTA(6,2),  XA(ISTDIM)
      REAL*8              YA(ISTDIM),  YOA(ISTDIM), YOAP(ISTDIM)
      REAL*8              Z,           Z0,          Z1
```

2.5 a6data: Subroutine a6data from library adas1xx

```
      SUBROUTINE A6DATA( DSFULL , INDXREF , TITLE , CAMETH , Z0    , Z ,
     &                   Z1     , NIGRP   , EMIN  , CIA    , NSHELA,
     &                   EIONA  , IZETAA  , NRGRP , CRA    , NRESOA,
     &                   ENERA  , WGHTA   , ICT   , XA     , YA    ,APA,
     &                   ITOUT  , TOA     , YOA   , YOAP   ,
     &                   ISTDIM , IREAD   , NA    , LA
     &     )
C-----------------------------------------------------------------------
C
C  ******************** FORTRAN77 SUBROUTINE A6DATA ******************
C
C  PURPOSE: TO REFRESH A DATA INDEX FROM AN ADAS106 ARCHIVE. READS
C           IN THE INDEX CODE A-ADAS, B-BURGESS AND THE THE REST OF
C           THE DATA AS APPROPRIATE.
C
C  CALLING PROGRAM:
C            ADAS106.FOR
C
C  INPUT:
C            (C*80) DSFULL   - THE USERS' CHOSEN ARCHIVE FILE NAME.
C            (I*4)  INDXREF  - THE INDEX NUMBER TO REFRESH FROM.
C            (C*4)  CAMETH   - THE TAG TO DISTINGUISH BETWEEN THE
C                              TWO TYPES OF ANALYSIS.
C                              A - ADAS, B- BURGESS
C            (I*4)  ISTDIM = ARRAY DIMENSIONS : MAX. NO OF VALUES THAT
C                            CAN BE READ IN
C            (I*4)  IREAD  = THE INPUT UNIT
C
C  OUTPUTS:
C            (C*40) TITLE    - THE INFORMATION LINE IN THE ARCHIVE
C                              FILE.
C            (R*8)  Z0     = NUCLEAR CHARGE OF ION
C            (R*8)  Z      = INITIAL ION CHARGE
C            (R*8)  Z1     = FINAL ION CHARGE
C            (I*4)  NIGRP  = NO. OF SHELL GROUPS
C            (R*8)  EMIN   = MINIMUM ENERGY (?)
C            (R*8)  CIA()  = SCALING PARAMETERS FOR SHELL GROUPS
C            (I*4)  NSHELA()=NO. OF ENTRIES FOR EACH SHELL GROUP
C            (I*4)  NA(,)  = SHELL GROUP DATA : N
C            (I*4)  LA(,)  = SHELL GROUP DATA : L
C            (R*8)  EIONA(,)=SHELL GROUP DATA : EION(RYD)
C            (I*4)  IZETAA(,)=SHELL GROUP DATA : IZETA
C            (I*4)  NRGRP  = NO. OF RESONANCE GROUPS
C            (R*8)  CRA()  = SCALING PARAMETERS FOR RESONANCE GROUPS
C            (I*4)  NRESOA()=NO. OF ENTRIES FOR EACH RESONANCE GROUP
C            (R*8)  ENERA(,)=RESONANCE GROUP DATA : ENERGY(RYD)
C            (R*8)  WGHTA(,)=RESONANCE GROUP DATA : WEIGHT
C            (R*8)  ICT    = NO. OF ENERGY / X-SECTION PAIRS
C            (R*8)  XA()   = X, THRESHOLD PARAMETER RELATIVE TO FIRST IONIS. POT.
C            (R*8)  YA()   = Q/(1-1/X), Q=CROSS-SECTION ?
C            (R*8)  APA()  = QEM/(1-1/X), QEM=APPROX. X-SECTION )
C            (R*4)  YPA()  = Q/QEM
C            (I*4)  ITOUT  = NO. OF TEMPS.
C            (R*8)  TOA()  = TEMP (KELVIN)
C            (R*8)  YOA()  = S, MAXWELL AVERAGED IONISATION RATE COEFF.(CM^3 S^-1)
C            (R*8)  YOAP() = SEM, APPROXIMATE RATE COEFF.
C
C  ROUTINES: NONE
C
```

```
C  AUTHOR:   WILLIAM OSBORN, TESSELLA SUPPORT SERVICES PLC., 28TH AUG 1996
C
C  VERSION 1.1                                          DATE: 28-08-96
C  MODIFIED: WILLIAM OSBORN
C - FIRST RELEASE
C
C-----------------------------------------------------------------------
      CHARACTER*4         CAMETH
      CHARACTER*80        DSFULL
      CHARACTER*40        TITLE
      INTEGER             ICT,          INDXREF,     IREAD,       ISTDIM
      INTEGER             ITOUT,        IZETAA(6,2), LA(6,2),     NA(6,2)
      INTEGER             NIGRP,        NRESOA(2),   NRGRP
      INTEGER             NSHELA(2)
      REAL*8              APA(ISTDIM), CIA(2),      CRA(2)
      REAL*8              EIONA(6,2),  EMIN,        ENERA(6,2)
      REAL*8              TOA(ISTDIM), WGHTA(6,2),  XA(ISTDIM)
      REAL*8              YA(ISTDIM),  YOA(ISTDIM), YOAP(ISTDIM)
      REAL*8              Z,           Z0,          Z1
```

2.6 a8afit: Subroutine a8afit from library adas1xx

```
       SUBROUTINE A8AFIT  ( Z0    , Z     , ZEFF  , IETYP ,
     &                      IXTYP , IND1  , IND2  , WI, WJ, EI, EJ,
     &                      IATYP , ACOEFF, IFTYP , IOTYP , IFOUT ,
     &                      IXMAX , ITMAX , EDAT  , XDAT  , TDAT  ,
     &                      IIORD , IIBTS , IIFPT , IIXOP , IIDIF ,
     &                      BXC   , BPXC  ,FXC1   ,
     &                      FXC2  , FXC3  , XKC   ,
     &                      XA    , YA    , APOMA ,
     &                      DIFOMA, TOA   , GOA   , APGOA , EXCRA ,
     &                      DEXCRA, GBARFA,
     &                      ICT   , ITOUT , S     , FIJ   , EIJ
     &                     )
C-----------------------------------------------------------------------
C  PURPOSE: TO ANALYSE ELECTRON IMPACT CROSS-SECTION DATA AND CONVERT TO
C  RATE COEFFICIENTS
C
C  VARIOUS FORMS OF DATA ENTRY ARE ALLOWED
C
C  DATA IS FITTED WITH APPROXIMATE FORMS TO AID INTERPOLATION DEPENDING
C  ON THE TRANSITION TYPE. THESE ARE
C                1. DIPOLE
C                2. NON-DIPOLE
C                3. SPIN CHANGE
C                4. OTHER
C
C  DATA ENTRY IS VIA CALL TO PANEL SUBROUTINE  SPFMA4E  AS FOLLOWS:
C
C  INPUT
C
C  OUTPUT
C      Z0     = NUCLEAR CHARGE OF ION
C      Z      = ION CHARGE
C      ZEFF   = ION CHARGE + 1
C      IETYP  = 1  LEVEL ENERGIES IN CM-1
C             = 2  LEVEL ENERGIES IN RYD
C      IXTYP  = 1  DIPOLE TRANSITION
C             = 2  NON-DIPOLE TRANSITION
C             = 3  SPIN CHANGE TRANSITION
C             = 4  OTHER
C      IND1   = LOWER LEVEL INDEX  (USER CHOICE)
C      IND2   = UPPER LEVEL INDEX  (USER CHOICE)
C      WI     = LOWER LEVEL STATISTICAL WEIGHT
C      WJ     = UPPER LEVEL STATISTICAL WEIGHT
C      EI     = LOWER LEVEL ENERGY (IN SELECTED UNITS)
C      EJ     = UPPER LEVEL ENERGY
C      IATYP  = 1  A-COEFFICIENT RETURNED
C             = 2  OSCILLATOR STRENGTH RETURNED
C             = 3  LINE STRENGTH RETURNED
C      ACOEFF = TRANSITION PROBABILITY (IN ABOVE FORM, DIPOLE CASE ONLY)
C      IFTYP  = 1 UPPER K**2 (RYD) FOR COLLISION ENERGY UNITS
C             = 2 LOWER K**2 (RYD)
C             = 3 UPPER (K/Z0)**2 (RYD)
C             = 4 X PARAMETER
C             = 5 UPPER (K/ZEFF)**2 (RYD)
C      IOTYP  = 1 EXCITATION CROSS-SECTION (PI*A0**2) RETURNED
C             = 2 DE-EXCITATION CROSS-SECTION (PI*A0**2) RETURNED
C             = 3 COLLISION STRENGTH RETURNED
C             = 4 SCALED COLLISION STRENGTH (Z**2*OMEGA) RETURNED
C      IFOUT  = 1 KELVIN FOR OUTPUT TEMPERATURE UNIT
```

```
C             = 2   EV   FOR OUTPUT TEMPERATURE UNIT
C             = 3 SCALED UNITS RETURNED  (TE(K)/Z1**2)
C             = 4 REDUCED UNITS RETURNED (KTE/EIJ)
C      IXMAX  = NUMBER OF ENERGY/X-SECT PAIRS ENTERED
C      ITMAX  = NUMBER OF OUTPUT TEMPERATURES ENTERED
C      EDAT(I)= INPUT ENERGIES (SELECTED UNITS)
C      XDAT(I)= INPUT X-SECTS  (SELECTED UNITS)
C      TDAT(I)= OUTPUT TEMPS.  (SELECTED UNITS)
C      IIORD  = 1  4-PT GAUSS-LAGUERRE QUADRATURE
C             = 2  8-PT GAUSS-LAGUERRE QUADRATURE
C             = 3 12-PT GAUSS-LAGUERRE QUADRATURE
C      IIGPH  = 0  NO X-SECT GRAPH TO BE PRODUCED
C             = 1     X-SECT GRAPH TO BE PRODUCED
C      IIGPG  = 0  NO GAMMA  GRAPH TO BE PRODUCED
C             = 1     GAMMA  GRAPH TO BE PRODUCED
C      IIBTS  = 0  BAD POINT OPTION OFF
C             = 1  BAD POINT OPTION ON
C      IIFPT  = 1  SELECT ONE POINT OPTIMISING
C             = 2  SELECT TWO POINT OPTIMISING
C      IIXOP  = 0  OPTIMISING OFF
C             = 1  OPTIMISING ON  (IF ALLOWED)
C      IIDIF  = 0  RATIO FITTING FOR DIPOLE X-SECT(ONLY WITH OPTIMISING)
C             = 1  DIFFERENCE FITTING FOR DIPOLE X-SECT
C
C
C  AUTHOR:   HUGH SUMMERS, UNIVERSITY OF STRATHCLYDE
C
C  DATE:     16-06-99 VERSION 1.1
C  MODIFIED: HUGH SUMMERS
C             -FIRST RELEASE
C
C  DATE:   07/07/2004 VERSION: 1.2
C  MODIFIED: ALLAN WHITEFORD
C - CHANGED PARAMS108 TO PARAMS
C
C  DATE:   15/05/2007 VERSION: 1.3
C  MODIFIED: ALLAN WHITEFORD
C - UPDATED COMMENTS AS PART OF SUBROUTINE
C                DOCUMENTATION PRODUCTION.
C
C-----------------------------------------------------------------------
       INCLUDE   'PARAMS'
C-----------------------------------------------------------------------
      INTEGER             IATYP,       ICT,          IETYP,       IFOUT
      INTEGER             IFTYP,       IIBTS,        IIDIF,       IIFPT
      INTEGER             IIORD,       IIXOP,        IND1,        IND2
      INTEGER             IOTYP,       ITMAX,        ITOUT,       IXMAX
      INTEGER             IXTYP
      REAL*8              ACOEFF,      APGOA(ISTDIM)
      REAL*8              APOMA(ISTDIM),            BPXC,        BXC
      REAL*8              DEXCRA(ISTDIM),           DIFOMA(ISTDIM)
      REAL*8              EDAT(ISTDIM),             EI,          EIJ
      REAL*8              EJ,          EXCRA(ISTDIM),           FIJ
      REAL*8              FXC1,        FXC2,         FXC3
      REAL*8              GBARFA(ISTDIM),           GOA(ISTDIM), S
      REAL*8              TDAT(ISTDIM),             TOA(ISTDIM), WI
      REAL*8              WJ,          XA(ISTDIM),   XDAT(ISTDIM)
      REAL*8              XKC,         YA(ISTDIM),   Z,           Z0
      REAL*8              ZEFF
```

2.7 a8amax: Subroutine a8amax from library adas1xx

```
       subroutine a8amax( ixtyp , ibpts , idiff ,
     &                    s     , eij   , wi    , wj    ,
     &                    bxc   , bpxc  , fxc1  ,
     &                    fxc2  , fxc3  , xkc   ,
     &                    ict   , xsa   , ysa   ,
     &                    itout , toa   , goa   , apgoa , excra ,
     &                    dexcra, gbarfa
     &                   )
c-----------------------------------------------------------------------
c
c  ******************** fortran77 subroutine a8amax ******************
c
c  purpose: to perform Maxwellian averages of collision strengths for
c           adas interpolative fit to neutrals
c
c  calling program:  adas108.for
c
c  input:
c            (i*4)  ixtyp    = type of transition (1=dipole,2=non-dipole
c                              non-spin change, 3=spin change, 4=null)
c            (i*4)  ibpts    = bad point switch (0=normal, 1=bad.pt)
c            (i*4)  idiff    = difference switch (0=ratio, 1=diff)
c            (r*8)  s        = line strength for dipole case
c            (r*8)  eij      = transition energy (rydberg)
c            (r*8)  wi       = lower level statistical weight
c            (r*8)  wj       = upper level statistical weight
c            (r*8)  bxc      = threshold form parameter
c            (r*8)  bpxc     = matching parameter
c            (r*8)  fxc1     = threshold form parameter
c            (r*8)  fxc2     = asymptotic form parameter
c            (r*8)  fxc3     = asymptotic form parameter
c            (i*4)  ict      = length of xsa and ysa value set
c            (r*8)  xsa()    = independent (energy) coord. for spline
c            (r*8)  ysa()    = dependent (coll. str.) coord. for spline
c  output:
c            (i*4)  itout    = length of toa, goa value set
c            (r*8)  toa()    = output temperatures (K)
c            (r*8)  goa()    = output upsilons
c            (r*8)  apgoa()  = output approximate form upsilons
c            (r*8)  excra()  = output excitation rate coefficients
c            (r*8)  dexcra() = output de-excitation rate coefficients
c            (r*8)  gbarfa() = output gbar*f coefficients
c
c
c  routines:
c             egasym adas generates asyptotic spline conditions
c       egspc adas    generates spline coefficients
c             elnfit    adas    obtains linearly interpolated value
c             efasym    adas    obtains spline interpolated value
c             a8gamg    adas    calculates incomplete gamma function
c             eei       adas    exponential integral exp(x)*e1(x)
c             ee2       adas    exponential integral exp(x)*e2(x)
c
c  author:   Hugh Summers, University of Strathclyde ext.4196
c
c
c  version 1.1                             date:     25/06/99
c  modified: Hugh Summers
c - first release
```

```
c
C  DATE:   07/07/2004 VERSION: 1.2
C  MODIFIED: ALLAN WHITEFORD
C - CHANGED PARAMS108 TO PARAMS
C
c-----------------------------------------------------------------------
       include  'PARAMS'
c-----------------------------------------------------------------------
      INTEGER             IBPTS,       ICT,         IDIFF,       ITOUT
      INTEGER             IXTYP
      REAL*8              APGOA(ISTDIM),            BPXC,        BXC
      REAL*8              DEXCRA(ISTDIM),           EIJ
      REAL*8              EXCRA(ISTDIM),            FXC1,        FXC2
      REAL*8              FXC3,        GBARFA(ISTDIM)
      REAL*8              GOA(ISTDIM), S,           TOA(ISTDIM), WI
      REAL*8              WJ,          XKC,         XSA(ISTDIM)
      REAL*8              YSA(ISTDIM)
```

2.8 a8data: Subroutine a8data from library adas1xx

```
       SUBROUTINE A8DATA( DSFULL , INDXREF , TITLE , CAMETH , Z0    ,
     &                    Z      , ZEFF    , INDL  , INDU   , EI    ,
     &                    EJ     , WI      , WJ    , ACOEFF , S     ,
     &                    FIJ    , EIJ     , IXTYP ,
     &                    BXC    , BPXC    , FXC1  ,
     &                    FXC2   , FXC3    , XKC   ,
     &                    IXOPS  , IBPTS   , IFPTS , IDIFF  , ICT   ,
     &                    ITOUT  , XA , YA , APOMA , DIFOMA , TOA   ,
     &                    GOA    , APGOA   , EXCRA , DEXCRA , GBARFA,
     &                    ISTDIM , IREAD   , IZ    , IZ0    , GF    ,
     &                    BBVAL  , BCVAL
     &                 )
C-----------------------------------------------------------------------
C
C  ********************* FORTRAN77 SUBROUTINE A8DATA ******************
C
C  PURPOSE: TO REFRESH A DATA INDEX FROM AN ADAS108 ARCHIVE. READS
C           IN THE INDEX CODE A-ADAS, B-BURGESS/SUMMERS AND THE THE REST
C           OF THE DATA AS APPROPRIATE.  9-KNOT BURGESS SPLINE VERSION
C
C  CALLING PROGRAM:
C            ADAS108.FOR
C
C  INPUT:
C            (C*80) DSFULL   - THE USERS' CHOSEN ARCHIVE FILE NAME.
C            (I*4)  INDXREF  - THE INDEX NUMBER TO REFRESH FROM.
C            (C*40) TITLE    - THE INFORMATION LINE IN THE ARCHIVE
C                              FILE.
C            (C*4)  CAMETH   - THE TAG TO DISTINGUISH BETWEEN THE
C                              TWO TYPES OF ANALYSIS.
C                              A - ADAS, B- BURGESS
C            (R*8)  GF       - THE WEIGHTED OSCILLATOR STRENGTH
C            (R*8)  BBVAL    - THE BURGESS SCALABLE PARAMETER B.
C            (R*8)  BCVAL    - THE BURGESS SCALABLE PARAMETER C.
C            (I*4)  ISTDIM = THE MAXIMUM ARRAY DIMENSION
C            (I*4)  IREAD  = THE INPUT UNIT
C
C  OUTPUTS:
C            (R*8)  Z0     = NUCLEAR CHARGE OF ION
C            (R*8)  Z      = ION CHARGE
C            (R*8)  ZEFF   = ION CHARGE + 1
C            (I*4)  INDL   = LOWER LEVEL INDEX  (USER CHOICE)
C            (I*4)  INDU   = UPPER LEVEL INDEX  (USER CHOICE)
C            (R*8)  WI     = LOWER LEVEL STATISTICAL WEIGHT
C            (R*8)  WJ     = UPPER LEVEL STATISTICAL WEIGHT
C            (R*8)  EI     = LOWER LEVEL ENERGY (IN SELECTED UNITS)
C            (R*8)  EJ     = UPPER LEVEL ENERGY
C            (R*8)  ACOEFF = TRANSITION PROBABILITY (IN ABOVE FORM,
C                            DIPOLE CASE ONLY)
C            (I*4)  IXTYP  = 1  DIPOLE TRANSITION
C                          = 2  NON-DIPOLE TRANSITION
C                          = 3  SPIN CHANGE TRANSITION
C                          = 4  OTHER
C            (I*4)  IBPTS  = 0  BAD POINT OPTION OFF
C                          = 1  BAD POINT OPTION ON
C            (I*4)  IFPTS  = 1  SELECT ONE POINT OPTIMISING
C                          = 2  SELECT TWO POINT OPTIMISING
C            (I*4)  IXOPS  = 0  OPTIMISING OFF
C                          = 1  OPTIMISING ON  (IF ALLOWED)
```

```
C            (I*4)  IDIFF  = 0  RATIO FITTING FOR DIPOLE X-SECT(ONLY
C                               WITH OPTIMISING)
C                           = 1  DIFFERENCE FITTING FOR DIPOLE X-SECT
C            (R*8)  S      = LINE STRENGTH
C            (R*8)  FIJ    = OSCILLATOR STRENGTH
C            (R*8)  EIJ    = TRANSITION ENERGY
C            (R*8)  BXC    = APPROX. FORM PARAMETER - LOW ENERGY
C            (R*8)  BPXC   = MATCHING PARAMETER
C            (R*8)  FXC1   = APPROX. FORM PARAMETER - LOW ENERGY
C            (R*8)  FXC2   = APPROX. FORM PARAMETER - HIGH ENERGY
C            (R*8)  FXC3   = APPROX. FORM PARAMETER - HIGH ENERGY
C            (R*8)  XKC    = SWITCHING X-VALUE BETWEEN LOW AND HIGH ENERGY.
C            (I*4)  ICT    = NUMBER OF X-SECTIONS
C            (I*4)  ITOUT  = NUMBER OF TEMPERATURES
C            (R*8)  XA     = ENERGY (PARAMETER X)
C            (R*8)  YA     = OMEGA (COLLISION STRENGTH)
C            (R*8)  APOMA  = APPROXIMATE OMEGA
C            (R*8)  DIFOMA = DIFFERENCE BETWEEN YA & APOMA
C            (R*8)  TOA    = TEMPERATURE SET
C            (R*8)  GOA    = GAMMA (EFFECTIVE COLLISION STRENGTHS)
C            (R*8)  APGOA  = APPROXIMATE GAMMA
C            (R*8)  EXCRA  = EXCITATION RATE COEFFICIENT
C            (R*8)  DEXCRA = DEEXCITATION RATE COEFFICIENT
C            (R*8)  GBARFA = G BAR FUNCTION
C            (I*4)  ISTDIM = THE MAXIMUM ARRAY DIMENSION
C            (I*4)  IREAD  = THE INPUT UNIT
C            (I*4)  IZ     = ION CHARGE (INTEGRAL)
C            (I*4)  IZ0    = NUCLEAR CHARGE (INTEGRAL)
C            (R*8)  GF     = GF-VALUE
C            (R*8)  BBVAL  = BURGESS B-VALUE
C            (R*8)  BCVAL  = BURGESS C-VALUE
C
C  ROUTINES: NONE
C
C  AUTHOR:   HUGH SUMMERS (UNIV.OF STRATHCLYDE) EXT.4196
C
C  DATE:  16/06/99 VERSION 1.1
C  MODIFIED: HUGH SUMMERS
C - FIRST RELEASE
C
C-----------------------------------------------------------------------
      CHARACTER*4         CAMETH
      CHARACTER*80        DSFULL
      CHARACTER*40        TITLE
      INTEGER             IBPTS,       ICT,          IDIFF,       IFPTS
      INTEGER             INDL,        INDU,         INDXREF,     IREAD
      INTEGER             ISTDIM,      ITOUT,        IXOPS,       IXTYP
      INTEGER             IZ,          IZ0
      REAL*8              ACOEFF,      APGOA(ISTDIM)
      REAL*8              APOMA(ISTDIM),            BBVAL,       BCVAL
      REAL*8              BPXC,        BXC,          DEXCRA(ISTDIM)
      REAL*8              DIFOMA(ISTDIM),           EI,          EIJ
      REAL*8              EJ,          EXCRA(ISTDIM),            FIJ
      REAL*8              FXC1,        FXC2,         FXC3
      REAL*8              GBARFA(ISTDIM),           GF
      REAL*8              GOA(ISTDIM), S,            TOA(ISTDIM), WI
      REAL*8              WJ,          XA(ISTDIM),   XKC
      REAL*8              YA(ISTDIM),  Z,            Z0,          ZEFF
```

2.9 a8gamg: Subroutine a8gamg from library adas1xx

```
       function a8gamg( a    , x )
c-----------------------------------------------------------------------
c
c  ******************** fortran77 function a8gamg ******************
c
c  purpose: to evaluate the incomplete gamma function gamma(a,x)
c           based on numerical recipes
c
c  calling program:  a8amax.for
c
c  input:
c            (r*8)  a      = parameter of p(a,x)
c            (r*8)  x      = parameter of p(a,x)
c  output:
c            (r*8)  a8gamg = incomplete gamma function gamma(a,x)
c                            (n.b. for x<0 principal value of
c                             logarithm is taken)
c
c
c  routines:
c             a8gser adas generates series expansion of gamma
c       a8gcf adas    generates continued fraction for gamma
c             a8gaml    adas    obtains log(gamma(a))
c
c  author:   Hugh Summers, University of Strathclyde ext.4196
c
c
c  version 1.1                               date:     27/06/99
c  modified: Hugh Summers
c - first release
c
c-----------------------------------------------------------------------
      REAL*8              A,           X
```

2.10 a8gaml: Subroutine a8gaml from library adas1xx

```
       function a8gaml( xx )
c-----------------------------------------------------------------------
c
c  ******************** fortran77 function a8gaml ******************
c
c  purpose: to evaluate log (gamma (xx)) for xx >0  - based on
c           numerical recipes
c
c  calling program:  a8gser.for , a8gcf
c
c  input:
c            (r*8)  xx      = parameter of gamma
c  output:
c            (r*8)  a8gaml = log gamma (xx)
c
c
c  routines:
c             none
c
c  author:   Hugh Summers, University of Strathclyde ext.4196
c
c
c  version 1.1                              date:     25/06/99
c  modified: Hugh Summers
c - first release
c
c-----------------------------------------------------------------------
      REAL*8              XX
```

2.11 a8gcf: Subroutine a8gcf from library adas1xx

```
       subroutine a8gcf( gammcf, a    , x    , gln )
c-----------------------------------------------------------------------
c
c  ******************** fortran77 subroutine a8gcf ******************
c
c  purpose: to evaluate the continued fraction expansion for the
c           incomplete gamma function gamma(a)*q(a,x) - based on
c           numerical recipes
c
c  calling program:  a8gamg.for
c
c  input:
c            (r*8)  a      = parameter of q(a,x)
c            (r*8)  x      = paramete of q(a,x)
c  output:
c            (r*8)  gammcf = incomplete gamma function gamma(a)*q(a,x)
c            (r*8)  gln    = ln(gamma(a))
c
c
c  routines:
c             none
c             a8gaml    adas    obtains log(gamma(a))
c
c  author:   Hugh Summers, University of Strathclyde ext.4196
c
c
c  version 1.1                             date:     25/06/99
c  modified: Hugh Summers
c - first release
c
c-----------------------------------------------------------------------
      REAL*8              A,           GAMMCF,      GLN,          X
```

2.12 a8gser: Subroutine a8gser from library adas1xx

```
       subroutine a8gser( gamser, a    , x    , gln )
c-----------------------------------------------------------------------
c
c  ******************** fortran77 function a8gser ******************
c
c  purpose: to evaluate the series expansion for the incomplete gamma
c           function p(a,x) - based on numerical recipes
c
c  calling program:  a8gamp.for
c
c  input:
c            (r*8)  a      = parameter of p(a,x)
c            (r*8)  x      = paramete of p(a,x)
c  output:
c            (r*8)  gamser = incomplete gamma function gamma(a,x)
c                            (n.b. for x<0 takes principal value of
c                             logarithm)
c            (r*8)  gln    = ln(gamma(a))
c
c
c  routines:
c             none
c             a8gaml    adas    obtains log(gamma(a))
c
c  author:   Hugh Summers, University of Strathclyde ext.4196
c
c
c  version 1.1                             date:     25/06/99
c  modified: Hugh Summers
c - first release
c
c-----------------------------------------------------------------------
      REAL*8              A,           GAMSER,      GLN,         X
```

2.13 a8optm: Subroutine a8optm from library adas1xx

```
       subroutine a8optm(itype,xa,oa,n,s,b0,bp0,f10,f20,f30,xk0,ifail)
c-----------------------------------------------------------------------
c
c  ********************* fortran77 subroutine a8optm ******************
c
c  purpose: to find the best approximate form parameters for neutral
c           atoms by varing the matching position.
c
c  calling program:  adas108.for
c
c  input:
c            (i*4)  itype    = type of transition (1=dipole,2=non-dipole
c                              non-spin change, 3=spin change, 4=null)
c            (r*8)  xa()     = x-parameters for cross-section
c            (r*8)  oa       = collisions strengths for transition
c            (i*4)  n        = no of collision strengths
c            (r*8)  s        = line strength if dipole transition
c            (i*4)  ifail    = failure code  on entry (ifail=0 two point
c                              fit, ifail=-1 one point fit)
c  output:
c            (r*8)  b0       = threshold form parameter
c            (r*8)  bp0      = matching parameter
c            (r*8)  f10      = threshold form parameter
c            (r*8)  f20      = asymptotic form parameter
c            (r*8)  f30      = asymptotic form parameter
c            (r*8)  xk0      = optimum matching x-value
c            (i*4)  ifail    = failure code  on exit
c                             (ifail=0 successful two point fit
c                              ifail=1 converted to one point fit)
c
c
c  routines:
c             a8slvf adas solves for asymptotic parms f2 and f3
c       a8slv2 adas    solves for the parms f1,f2,f3,b
c
c  author:   Hugh Summers, University of Strathclyde ext.4196
c
c
c  version 1.1                              date:     19/07/99
c  modified: Hugh Summers
c - first release
c
c-----------------------------------------------------------------------
      INTEGER             IFAIL,       ITYPE,       N
      REAL*8              B0,          BP0,         F10,         F20
      REAL*8              F30,         OA(N),       S,           XA(N)
      REAL*8              XK0
```

2.14 a8slv2: Subroutine a8slv2 from library adas1xx

```
       subroutine a8slv2( itype, x0  , sig0 , xk  , sigk , xn  , sign ,
     &                    s    , f1  , f2   , f3  , b    , bp  , ifail
     &                  )
c-----------------------------------------------------------------------
c
c  ******************** fortran77 subroutine a8slv2 ******************
c
c  purpose: to find the approximate form parameters for a neutral atoms
c
c  calling program:  adas108.for
c
c  input:
c            (i*4)  itype    = type of transition (1=dipole,2=non-dipole
c                              non-spin change, 3=spin change, 4=null)
c            (r*8)  x0       = x-parameter at first energy point n
c            (r*8)  sig0     = collisions strength at first energy pt. n
c            (r*8)  xk       = x-parameter at matching point k
c            (r*8)  sigk     = collisions strength at matching point k
c            (r*8)  xn       = x-parameter at last energy point n
c            (r*8)  sign     = collisions strength at last energy pt. n
c            (r*8)  s        = line strength for type 1 case
c            (i*4)  ifail    = failure code  on entry (ifail=0 two point
c                              fit, ifail=-1 one point fit)
c  output:
c            (r*8)  f1       = threshold form parameter
c            (r*8)  f2       = asymptotic form parameter
c            (r*8)  f3       = asymptotic form parameter
c            (r*8)  b        = threshold form parameter
c            (r*8)  bp       = matching parameter
c            (i*4)  ifail    = failure code  on exit
c                             (ifail=0 successful two point fit
c                              ifail=1 converted to one point fit)
c
c
c  routines:
c             a8slvf adas solves for asymptotic parms f2 and f3
c
c  author:   Hugh Summers, University of Strathclyde ext.4196
c
c
c  version 1.1                               date:     16/06/99
c  modified: Hugh Summers
c - first release
c
c-----------------------------------------------------------------------
      INTEGER             IFAIL,       ITYPE
      REAL*8              B,           BP,          F1,          F2
      REAL*8              F3,          S,           SIG0,        SIGK
      REAL*8              SIGN,        X0,          XK,          XN
```

2.15 a8slvf: Subroutine a8slvf from library adas1xx

```
       subroutine a8slvf(itype,xk,sigk,xn,sign,s,f2,f3,ifail)
c-----------------------------------------------------------------------
c
c  ******************** fortran77 subroutine a8slvf ******************
c
c  purpose: to find the approximate form parameters f2 and f3 for
c           neutrals
c
c  calling program:  adas108.for
c
c  input:
c            (i*4)  itype    = type of transition (1=dipole,2=non-dipole
c                              non-spin change, 3=spin change, 4=null)
c            (r*8)  xk       = x-parameter at matching point k
c            (r*8)  sigk     = collisions strength at matching point k
c            (r*8)  xn       = x-parameter at last energy point n
c            (r*8)  sign     = collisions strength at last energy pt. n
c            (r*8)  s        = line strength for type 1 case
c            (i*4)  ifail    = failure code  on entry (ifail=0 two point
c                              fit, ifail=-1 one point fit)
c  output:
c            (r*8)  f2       = asymptotic form parameter
c            (r*8)  f3       = asymptotic form parameter
c            (i*4)  ifail    = failure code  on exit
c                             (ifail=0 successful two point fit
c                              ifail=1 converted to one point fit)
c
c
c  routines: none
c
c  author:   Hugh Summers, University of Strathclyde ext.4196
c
c
c  version 1.1                              date:     15/06/99
c  modified: Hugh Summers
c - first release
c
c-----------------------------------------------------------------------
      INTEGER             IFAIL,       ITYPE
      REAL*8              F2,          F3,          S,           SIGK
      REAL*8              SIGN,        XK,          XN
```

2.16 axetrd: Subroutine axetrd from library adas1xx

```
       FUNCTION AXETRD( KTYPE , E , T ,  C)
C-----------------------------------------------------------------------
C
C  ******************** FORTRAN 77 FUNCTION: AXETRD *********************
C
C  PURPOSE:  TO CALCULATE THE REDUCED ENERGY FOR FOUR TYPES OF
C            TRANSITION
C
C  CALLING PROGRAM: VARIOUS ADAS101 CODES
C
C  FUNCTION:
C
C  INPUT:    (R*8)    E      =   TRANSITION ENERGY (Eij)
C            (R*8)    T      =   COLLIDING ELECTRON ENERGY AFTER
C                                 EXCITATION (Ej)
C            (R*8)    C      =   ADJUSTABLE SCALING PARAMETER
C            (R*8)    TL     =   Ej/Eij OR kTe/Eij
C            (I)      KTYPE  =   TRANSITION TYPE
C                                 1 ELECTRIC DIPOLE
C                                 2 NON ELECTRIC DIPOLE
C                                 3 SPIN CHANGE
C                                 4 OTHER
C
C  COMMON:
C        /BURG/
C            (L*4)    LUPSIL = .TRUE.  (UPSILON FITTING)
C                              .FALSE. (OMEGA FITTING  )
C
C
C  OUTPUT:   (R*8)    AXETRD  =   REDUCED ENERGY OR TEMPERATURE
C
C
C  ROUTINES: NONE
C
C  WRITTEN:   CONVERSION OF ETRED BY A.LANZAFAME & D.H.BROOKS BY
C             HUGH P. SUMMERS, UNIVERSITY OF STRATHCLYDE
C             TEL. 0141-553-4196
C
C  DATE:     24/11/96 VERSION 1.1
C
C  MODIFICATION HISTORY:
C
C  VERSION: 1.1   HUGH SUMMERS  24/11/96
C  MODIFIED:      FIRST RELEASE
C
C-----------------------------------------------------------------------
      INTEGER             KTYPE
      REAL*8              C,           E,           T
```

2.17 axetrdv: Subroutine axetrdv from library adas1xx

```
       FUNCTION AXETRDV( IXTYP , EIJ   , DGEX  , C )
C-----------------------------------------------------------------------
C
C  ******************** FORTRAN 77 FUNCTION: AXETRDV ******************
C
C  PURPOSE:  TO CALCULATE THE ELECTRON ENERGY FROM THE REDUCED ENERGY
C            FOR FOUR TYPES OF TRANSITION
C
C  CALLING PROGRAM:
C
C  FUNCTION:
C
C  INPUT:    (R*8)    EIJ    =   TRANSITION ENERGY (Eij)
C            (R*8)    DGEX   =   REDUCED ENERGY
C            (R*8)    C      =   ADJUSTABLE SCALING PARAMETER
C            (R*8)    TL     =   Ej/Eij
C            (I)      IXTYPE =   TRANSITION TYPE
C                                1 ELECTRIC DIPOLE
C                                2 NON ELECTRIC DIPOLE
C                                3 SPIN CHANGE
C                                4 OTHER
C
C  OUTPUT:   (R*8)    AXETRDV=   ELECTRON ENERGY
C
C
C  ROUTINES: NONE
C
C  WRITTEN:   CONVERSION OF ETREDINV BY A.LANZAFAME & D.H.BROOKS BY
C             HUGH P. SUMMERS, UNIVERSITY OF STRATHCLYDE
C             TEL. 0141-553-4196
C
C  DATE:     24/11/96 VERSION 1.1
C
C  MODIFICATION HISTORY:
C
C  VERSION: 1.1   HUGH SUMMERS  24/11/96
C  MODIFIED:      FIRST RELEASE
C
C-----------------------------------------------------------------------
      INTEGER             IXTYP
      REAL*8              C,           DGEX,        EIJ
```

2.18 axiups: Subroutine axiups from library adas1xx

```
      SUBROUTINE AXIUPS(IT,C,E,P,NT,T,U)
C-----------------------------------------------------------------------
C  ******************  FORTRAN77 SUBROUTINE: AXIUPS  *******************
C
C  PURPOSE: TO CALCULATE A SET OF USPILONS BY INTERPOLATION OF THE
C           BURGESS FIVE-POINT SPLINE
C
C  INPUT:
C    IT  TRANSITION TYPE
C    C   UPSILON SCALING PARAMETER
C    E   EXCITATION ENERGY
C    P   KNOTS
C    NT  NUMBER OF TEMPERATURE POINTS
C    T   TEMPERATURE
C
C  OUTPUT:
C    U   UPSILON
C
C  ROUTINES:
C    AXOUPS - TO CALCULATE THE MAXWELL AVERAGED UPSILONS
C
C  AUTHOR:  HUGH P. SUMMERS, UNIVERSITY OF STRATHCLYDE
C           TEL. 0141-553-4196
C
C  DATE:    24 NOVEMBER 1996
C
C  MODIFICATION HISTORY:
C
C  VERSION: 1.1   HUGH SUMMERS  24/11/96
C  MODIFIED:      FIRST RELEASE.
C
C-----------------------------------------------------------------------
      INTEGER             IT,          NT
      REAL*8              C,           E,           P(5),        T(NT)
      REAL*8              U(NT)
```

2.19 axltsq: Subroutine axltsq from library adas1xx

```
      SUBROUTINE AXLTSQ(IT,C,E12,GF,N,T,U,B,RMS)
C-----------------------------------------------------------------------
C  ******************** FORTRAN77 SUBROUTINE: AXLTSQ ******************
C
C  PURPOSE: TO PERFORM FIVE POINT SPLINE FIT TO REDUCED OMEGAS OR
C           UPSILONS
C
C  INPUT:
C           E12  TRANSITION ENERGY  (RYD)
C           GF   GF
C           IT   TRANSITION TYPE
C           T    ENERGY OR TEMPERATURE
C           U    OMEGA OR UPSILON
C           N    NUMBER OF DATA POINTS
C           C    C PARAMETER
C
C  COMMON:
C         /BURG/
C           LUPSIL    = .TRUE.  (UPSILON FITTING)
C                       .FALSE. (OMEGA FITTING  )
C
C  OUTPUT:
C           B    KNOTS VALUES
C
C  LOCAL VARIABLES/CONSTANTS:
C
C           A(,)
C           V
C           W
C           Y()
C           XX
C           YY
C
C  CALLS:
C          SPLS     - CALCULATE CUBIC SPLINE FIT COEFFICIENTS
C          AXETRD   - CALCULATE REDUCED ENERGIES OR TEMPERATURES
C          AXOURD   - CALCULATE REDUCED OMEGAS OR UPSILONS
C          MATIN1   - INVERT MATRIX TO GET KNOT POINTS
C          ONE      - GET KNOT POINTS IF ONLY ONE DATA POINT
C          TWO      - GET KNOT POINTS IF ONLY TWO DATA POINTS
C          THREE    - GET KNOT POINTS IF ONLY THREE DATA POINTS
C          FOUR     - GET KNOT POINTS IF ONLY FOUR DATA POINTS
C
C  DATE: 24-11-96 VERSION 1.1
C  WRITTEN: CONVERSION OF LSTSQ BY A.LANZAFAME & D.H.BROOKS BY
C           HUGH P. SUMMERS, UNIVERSITY OF STRATHCLYDE
C           TEL. 0141-553-4196
C
C  DATE:    24 NOVEMBER 1996
C
C  MODIFICATION HISTORY:
C
C  VERSION: 1.1   HUGH SUMMERS  24/11/96
C  MODIFIED:      FIRST RELEASE.
C-----------------------------------------------------------------------
      INTEGER             IT,          N
      REAL*8              B(5),        C,           E12,         GF
      REAL*8              RMS,         T(N),        U(N)
```

2.20 axomup: Subroutine axomup from library adas1xx

```
      FUNCTION AXOMUP(IT,E,T,U,C)
C-----------------------------------------------------------------------
C
C  ******************** FORTRAN77 SUBROUTINE: AXOMUP *********************
C
C  PURPOSE:
C        TO INTERPOLATE OMEGAS OR UPSILONS FOR DIFFERENT TRANSITIONS
C
C  CALLING PROGRAM:
C        OMEUPS
C
C  INPUT:
C        (I*4)    IT - TRANSITION TYPE
C        (R*8)    E - EXCITATION ENRGY
C        (R*8)    T - SCALED ENERGY VALUE OF QUADRATURE FIXED POINTS
C        (R*8)    U - SPLINE FIT TO THE KNOT POINTS AND
C                     REDUCED ENERGIES OR TEMPERATURES
C        (R*8)    C - SCALABLE PARAMETER
C
C  COMMON:
C     /BURG/
C        (L*4)    LUPSIL = .TRUE.  (UPSILON FITTING)
C                          .FALSE. (OMEGA FITTING  )
C
C  OUTPUT:
C        (R*8) AXOMUP - THE UPSILONS
C
C
C  WRITTEN:   CONVERSION OF OMUP BY A.LANZAFAME & D.H.BROOKS BY
C             HUGH P. SUMMERS, UNIVERSITY OF STRATHCLYDE
C             TEL. 0141-553-4196
C
C  DATE:     24/11/96 VERSION 1.1
C  MODIFIED:  FIRST RELEASE
C
C  DATE:   18/05/99 VERSION 1.2
C  AUTHOR: HUGH SUMMERS, UNIVERSITY OF STRATHCLYDE
C          PREVENTED CHANGE OF INPUT SUBROUTINE PARAMETER T ON RETURN
C
C-----------------------------------------------------------------------
      INTEGER             IT
      REAL*8              C,           E,           T,           U
```

2.21 axoups: Subroutine axoups from library adas1xx

```
      FUNCTION AXOUPS(IT,E,C,P,X)
C-----------------------------------------------------------------------
C
C  ******************** FORTRAN77 SUBROUTINE:AXOUPS*********************
C
C  PURPOSE:
C        TO CALCULATE UPSILONS
C
C  CALLING PROGRAM:
C        MAXWELL
C
C  INPUT:
C        (I)     IT - TRANSITION TYPE
C        (R*8)    E - EXCITATION ENRGY
C        (R*8)    C - SCALABLE PARAMETER
C        (R*8)    P - KNOT POINTS
C        (R*8)    X - COLLIDING ELECTRON ENERGY AFTER
C                     EXCITATION
C  OUTPUT:
C        (R*8) AXOUPS - THE UPSILONS
C
C  ROUTINES:
C
C  WRITTEN:   CONVERSION OF OMEUPS BY A.LANZAFAME & D.H.BROOKS BY
C             HUGH P. SUMMERS, UNIVERSITY OF STRATHCLYDE
C             TEL. 0141-553-4196
C
C  DATE:     24/11/96 VERSION 1.1
C
C  MODIFICATION HISTORY:
C
C  VERSION: 1.1   HUGH SUMMERS  24/11/96
C  MODIFIED:      FIRST RELEASE
C
C
C-----------------------------------------------------------------------
      INTEGER             IT
      REAL*8              C,           E,           P(5),        X
```

2.22 axourd: Subroutine axourd from library adas1xx

```
       FUNCTION AXOURD( KTYPE, EIJ  , EJ   , OMUP , C    )
C-----------------------------------------------------------------------
C
C  ******************** FORTRAN 77 FUNCTION: AXOURD *********************
C
C  PURPOSE:  TO CALCULATE THE REDUCED COLLISION STRENGTH OR REDUCED
C            UPSILON AS A FUNCTION OF Ej/Eij  OR kTE/EIJ FOR FOUR
C            TYPES OF TRANSITION
C
C  CALLING PROGRAM: VARIOUS ADAS101  AND ADAS102 ROUTINES
C
C  FUNCTION:
C
C  INPUT:    (R*8)    EIJ    =   TRANSITION ENERGY (Eij)
C            (R*8)    EJ     =   COLLIDING ELECTRON ENERGY AFTER
C                                  EXCITATION (Ej)
C            (R*8)    C      =   ADJUSTABLE SCALING PARAMETER
C            (R*8)    ETR     =  Ej/Eij OR kTe/Eij
C            (R*8)    OMUP   =   COLLISION STRENGTH OR UPSILON
C                                  AS A FUNCTION
C                                  OF ETR
C            (I)      KTYPE  =   TRANSITION TYPE
C                                  1 ELECTRIC DIPOLE
C                                  2 NON ELECTRIC DIPOLE
C                                  3 SPIN CHANGE
C                                  4 OTHER
C
C  COMMON:
C        /BURG/
C            (L*4)    LUPSIL = .TRUE.  (UPSILON FITTING)
C                              .FALSE. (OMEGA FITTING  )
C
C  OUTPUT:   (R*8)    AXOURD =   REDUCED COLLISION STRENGTH OR UPSILON
C
C
C  ROUTINES: NONE
C
C  WRITTEN:   CONVERSION OF OURED BY A.LANZAFAME & D.H.BROOKS BY
C             HUGH P. SUMMERS, UNIVERSITY OF STRATHCLYDE
C             TEL. 0141-553-4196
C
C  DATE:     24/11/96 VERSION 1.1
C
C  MODIFICATION HISTORY:
C
C  VERSION: 1.1   HUGH SUMMERS  24/11/96
C  MODIFIED:      FIRST RELEASE.
C
C-----------------------------------------------------------------------
      INTEGER             KTYPE
      REAL*8              C,           EIJ,         EJ,          OMUP
```

2.23 axourdv: Subroutine axourdv from library adas1xx

```
       FUNCTION AXOURDV( IXTYP , EIJ   , EJ  , DGEY  , C )
C-----------------------------------------------------------------------
C
C  ******************** FORTRAN 77 FUNCTION: AXOURDV *****************
C
C  PURPOSE:  TO CALCULATE THE COLLISION STRENGTH OR UPSILON FROM THE
C            REDUCED COLLISION STRENGTH OR REDUCED UPSILON
C            FOR FOUR TYPES OF TRANSITION
C
C  CALLING PROGRAM:
C
C  FUNCTION:
C
C  INPUT:    (R*8)    EIJ    =   TRANSITION ENERGY (Eij)
C            (R*8)    EJ     =   COLLIDING ELECTRON ENERGY AFTER
C                                 EXCITATION (Ej)
C            (R*8)    C      =   ADJUSTABLE SCALING PARAMETER
C            (R*8)    ETR    =   Ej/Eij
C            (I)      IXTYPE =   TRANSITION TYPE
C                                 1 ELECTRIC DIPOLE
C                                 2 NON ELECTRIC DIPOLE
C                                 3 SPIN CHANGE
C                                 4 OTHER
C            (R*8)    DGEY   =   REDUCED COLLISION STRENGTH
C
C  COMMON:
C        /BURG/
C            (L*4)    LUPSIL = .TRUE.  (UPSILON FITTING)
C                              .FALSE. (OMEGA FITTING  )
C
C  OUTPUT:   (R*8)    AXOURD =   REDUCED COLLISION STRENGTH OR UPSILON
C
C
C  ROUTINES: NONE
C
C  WRITTEN:   CONVERSION OF OUREDINV BY A.LANZAFAME & D.H.BROOKS BY
C             HUGH P. SUMMERS, UNIVERSITY OF STRATHCLYDE
C             TEL. 0141-553-4196
C
C  DATE:     24/11/96 VERSION 1.1
C
C  MODIFICATION HISTORY:
C
C  VERSION: 1.1   HUGH SUMMERS  24/11/96
C  MODIFIED:      FIRST RELEASE.
C
C-----------------------------------------------------------------------
      INTEGER             IXTYP
      REAL*8              C,           DGEY,        EIJ,         EJ
```

2.24 axwups: Subroutine axwups from library adas1xx

```
      SUBROUTINE AXWUPS( ELO   , EHI   , DELTAE, KTYPE , GF    ,
     &                   CPAR  , YKN   , I1    , I2    , NELEC ,
     &                   NCHAR , NTEMP , T     , UPS   , DSFULL,
     &                   ISTDIM, NET   , ENTE  , OMUP  , IWRITE,
     &                   INDIM , WI    , WJ
     &                   )
C-----------------------------------------------------------------------
C
C  ******************** FORTRAN77 SUBROUTINE:AXWUPS *****************
C
C  PURPOSE: TO WRITE DATA TO AN OLD/NEW ARCHIVE IN BURGESS FORMAT
C
C  INPUT:
C        (R*8) ELO    - LOWER LEVEL ENERGY
C        (R*8) EHI    - UPPER LEVEL ENERGY
C        (R*8) DELTAE - TRANSITION ENERGY
C        (I)   KTYPE  - TRANSITION TYPE
C        (R*8) GF     - WEIGHTED OSCILLATOR STRENGTH
C        (R*8) CPAR   - SCALABLE PARAMETER
C        (R*8) YKN    - KNOT POINTS
C        (I)   I1     - LOWER LEVEL INDEX
C        (I)   I2     - UPPER LEVEL INDEX
C        (R*8) NELEC  - NUMBER OF ELECTRONS
C        (R*8) NCHAR  - NUCLEAR CHARGE
C        (I)   NTEMP  - NUMBER OF TEMPERATURE POINTS
C        (R*8) T      - TEMPERATURES
C        (R*8) UPS    - UPSILONS
C        (C*80)DSFULL - ARCHIVE FILE NAME
C        (I)   ISTDIM - MAXIMUM INPUT ARRAY DIMENSIONS
C        (I)   NET    - NUMBER OF ENERGY POINTS
C        (R*8) ENTE   - ENRGIES
C        (R*8) OMUP   - OMEGAS
C        (I)   IWRITE - OUTPUT UNIT NUMBER
C        (I)   INDIM  - MAXIMUM OUTPUT ARRAY DIMENSION
C        (R*8) WI     - LOWER LEVEL STATISTICAL WEIGHT
C        (R*8) WJ     - UPPER LEVEL STATISTICAL WEIGHT
C
C  OUTPUT:
C
C  DATA:
C         CIARR(500)*80- 500 IS THE CURRENT LIMIT ON INDEXES
C  ROUTINES:
C         NONE
C
C
C  WRITTEN:   CONVERSION OF WUPSILON BY A.LANZAFAME & D.H.BROOKS BY
C             HUGH P. SUMMERS, UNIVERSITY OF STRATHCLYDE
C             TEL. 0141-553-4196
C
C  DATE:     24/11/96 VERSION 1.1
C  MODIFIED: HUGH SUMMERS  24/11/96
C   -  FIRST RELEASE.
C
C  DATE: 13/05/99 VERSION 1.2
C  MODIFIED: HUGH SUMMERS
C - CORRECTED CONFUSION ABOUT NCHAR AND NELEC IN COMMENTS
C                 INSERTED LEADING DECIMAL IN OUTPUT FLOATING FORMATS
C
C-----------------------------------------------------------------------
```

```
CHARACTER*80        DSFULL
INTEGER             I1,          I2,          INDIM,       ISTDIM
INTEGER             IWRITE,      KTYPE,       NET,         NTEMP
REAL*8              CPAR,        DELTAE,      EHI,         ELO
REAL*8              ENTE(ISTDIM),             GF,          NCHAR
REAL*8              NELEC,       OMUP(ISTDIM)
REAL*8              T(ISTDIM),   UPS(INDIM),  WI,          WJ
REAL*8              YKN(5)
```

2.25 burgfs: Subroutine burgfs from library adas1xx

```
       SUBROUTINE BURGFS(MAXT,TEA,ALFO,ALFGF,Z1,N0,V0,
     &NI,LI,WI,NJ,LJ,WJ,EIJ,F,EDISPG,SCALEG,PHFRAC,CORFAC,IOPT)
       IMPLICIT REAL*8(A-H,O-Z)
C-----------------------------------------------------------------------
C  PURPOSE: TO PROVIDE BURGESS GENERAL FORMULA RESULTS AT A SERIES OF
C  TEMPERATURES, AND ALSO TO PRODUCE BURGESS GENERAL PROGRAM RESULTS AT
C  ZERO DENSITY AT THE SAME TEMPERATURES.
C
C  THE LATTER ARE ADJUSTED TO EQUAL THE GENERAL FORMULA RESULTS AS FAR
C  AS POSSIBLE BY MODIFICATION OF BETHE CORRECTIONS VIA A SINGLE
C  SCALING PARAMETER CORFAC AND BY ADJUSTMENT OF THE PROPORTION OF THE
C  LOWEST LEVEL CAPTURE ALLOWED BASED ON AVAILABLE PHASE SPACE OF
C  OCCUPIED SHELLS ARGUMENTS VIA THE PARAMETER PHFRAC. THE CORRECTION
C  FACTORS USED IN THE GENERAL PROGRAM ARE OBTAINED BY ADJUSTMENT OF
C  STANDARD SETS FOR SPECIFIC TYPES OF TRANSITION.  THE ADJUSTMENT IS
C
C    (NEW COR(J))=EXP(-CORFAC/(L**DF+0.5))*(STANDARD COR(J)
C  THE STANDARD COR'S ARE AS FOLLOWS:
C   TYPE    TRANSITION                      COR'S                      DF
C     1   NI=1,NJ>=2,LJ=LI+1:       0.05,0.30,0.50,0.90              2.0
C     2   NI=2,NJ=3,LJ=LI+1:        0.01,0.02,0.20,0.40,0.70,0.90    1.0
C     3   NI=2,NJ=3,LJ=LI-1:        0.01,0.01,0.01,0.08,0.30,0.70    1.0
C     4   NJ-NI=0, LJ=LI+1 :        0.30,0.35,0.40,0.45,0.70,0.90    0.5
C     5   NJ-NI=0, LJ=LI-1 :        0.30,0.35,0.40,0.45,0.70,0.90    0.5
C     6   NJ-NI>0, LJ=LI+1 :        0.01,0.02,0.20,0.40,0.70,0.90    1.0
C     7   NJ-NI>0, LJ=LI-1 :        0.01,0.01,0.01,0.08,0.30,0.70    1.0
C  ********** H.P. SUMMERS, JET          11 JUNE 1987  ****************
C  ********** W. DICKSON, JET   CORR.    14 DEC  1987  ****************
C  INPUT
C      MAXT=NUMBER OF TEMPERATURES
C      TEA(I)=ELECTRON TEMPERATURES (K)
C      Z1=RECOMBINING ION CHARGE
C      N0=LOWEST ACCESSIBLE N-SHELL BY RECOMBINATION
C      V0=EFFECTIVE PRINCIPAL QUANTUM NUMBER OF LOWEST ACCESSIBLE SHELL
C      NI=LOWER PRINCIPAL QUANTUM NUMBER OF PARENT TRANSITION
C      LI=LOWER ANGULAR QUANTUM NUMBER OF PARENT TRNASITION.
C      WI=LOWER PARENT STATE STATISTICAL WEIGHT.
C      NJ=UPPER PRINCIPAL QUANTUM NUMBER OF PARENT TRANSITION
C      LJ=UPPER ANGULAR QUANTUM NUMBER OF PARENT TRNASITION.
C      WJ=UPPER PARENT STATE STATISTICAL WEIGHT.
C      EIJ=PARENT TRANSITION ENERGY (RYD)
C      FIJ=ABSORPTION OSCILLATOR STRENGTH OF PARENT TRANSITION
C      EDISPG=UNIFORM ENERGY DISPLACEMENT FOR GENERAL FORMULA
C      SCALEG=UNIFORM SCALING OF GENERAL FORMULA
C      PHFRAC=INITIAL ESTIMATE OF PHASE SPACE FACTOR
C      CORFAC=INITIAL ESTIMATE OF BETHE CORRECTION SCALER
C      IOPT=1  RETURNS GENERAL FORMULA RESULTS ONLY.
C          =2  RETURNS GENERAL FORMULA AND PROGRAM RESULTS WITH THE
C              THE INITIAL VALUE OF CORFAC USED.
C          =3  RETURNS GENERAL FORMULA AND PROGRAM RESULTS WITH
C              CORFAC ADJUSTED TO GIVE AGREEMENT BETWEEN THE TWO.
C  OUTPUT
C      ALFO(I)=GENERAL PROGRAM DIELECTRONIC COEFFICIENTS (CM+3 SEC-1)
C      ALFGFA(I)=GENERAL FORMULA DIELECTRONIC COEFFICIENTS
C      PHFRAC=REVISED PHASE SPACE FACTOR
C      CORFAC=REVISED BETHE CORRECTION SCALER
C-----------------------------------------------------------------------
C IDL-UNIX CONVERSION:
C
```

```
C  VERSION: 1.1                                          DATE: 01/10/96
C  MODIFIED:  WILLIAM OSBORN
C           - FIRST WRITTEN. NO CHANGES.
C
C  VERSION: 1.2                                          DATE: 15/05/07
C  MODIFIED:  Allan Whiteford
C           - Updated comments as part of subroutine
C             documentation production.
C
C-----------------------------------------------------------------------
      INTEGER             IOPT,         LI,           LJ,           MAXT
      INTEGER             N0,           NI,           NJ
      REAL*8              ALFGF(10),    ALFO(10),     CORFAC,       EDISPG
      REAL*8              EIJ,          F,            PHFRAC,       SCALEG
      REAL*8              TEA(10),      V0,           WI,           WJ
      REAL*8              Z1
```

2.26 dnsort: Subroutine dnsort from library adas1xx

```
       SUBROUTINE DNSORT(XA,YA,N)
       IMPLICIT REAL*8(A-H,O-Z)
C-----------------------------------------------------------------------
C  PURPOSE: TO SORT AN ARRAY SO THAT FIRST INPUT IS IN INCREASING ORDER
C
C  N.B.  INPUT VALUES ARE ALTERED BY THIS ROUTINE !!!!
C
C  INPUT
C      XA(I)=X-VALUES
C      YA(I)=Y-VALUES
C      N=NUMBER OF VALUES
C  OUTPUT
C      XA(I)=SORTED X-VALUES
C      YA(I)=SORTED Y-VALUES
C
C
C  *********                                              **************
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 07-11-96
C MODIFIED: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C               - PUT UNDER S.C.C.S. CONTROL
C
C VERSION: 1.2                          DATE: 15-05-07
C MODIFIED: Allan Whiteford
C               - Updated comments as part of subroutine
C                 documentation production.
C
C-----------------------------------------------------------------------
      INTEGER             N
      REAL*8              XA(10),      YA(10)
```

2.27 eight9: Subroutine eight9 from library adas1xx

```
      SUBROUTINE EIGHT9(A,P)
C-----------------------------------------------------------------------
C
C  ******************** FORTRAN 77SUBROUTINE: EIGHT9 ******************
C
C  PURPOSE:
C        TO CALCULATE THE BURGESS KNOT POINTS
C
C  CALLING PROGRAM:
C        LSTSQ9
C
C  INPUT:
C        (R*8) A - MATRIX CONTAINING REDUCED PARAMETERS
C  OUTPUT:
C        (R*8) P - THE BURGESS KNOT POINTS
C
C  ROUTINES:
C        MATIN1
C
C  DATE:   25/05/99 VERSION 1.1
C
C  AUTHOR: HUGH SUMMERS, UNIVERSITY OF STRATHCLYDE
C
C-----------------------------------------------------------------------
      REAL*8              A(9,9),      P(9)
```

2.28 etred: Subroutine etred from library adas1xx

```
       FUNCTION ETRED( KTYPE , E , T ,  C)
C-----------------------------------------------------------------------
C
C  ******************** FORTRAN 77 FUNCTION: ETRED *********************
C
C  PURPOSE:  TO CALCULATE THE REDUCED ENERGY FOR FOUR TYPES OF
C            TRANSITION
C
C  CALLING PROGRAM: VARIOUS ADAS101 CODES
C
C  FUNCTION:
C
C  INPUT:    (R*8)    E      =   TRANSITION ENERGY (Eij)
C            (R*8)    T      =   COLLIDING ELECTRON ENERGY AFTER
C                                 EXCITATION (Ej)
C            (R*8)    C      =   ADJUSTABLE SCALING PARAMETER
C            (R*8)    TL     =   Ej/Eij
C            (I)      KTYPE  =   TRANSITION TYPE
C                                 1 ELECTRIC DIPOLE
C                                 2 NON ELECTRIC DIPOLE
C                                 3 SPIN CHANGE
C                                 4 OTHER
C
C  OUTPUT:   (R*8)    ETRED  =   REDUCED ENERGY
C
C
C  ROUTINES: NONE
C
C  DATE:     21/06/95 VERSION 1.1
C  AUTHOR:   A.LANZAFAME (UNIV. OF STRATHCLYDE) & D.H.BROOKS
C
C-----------------------------------------------------------------------
      INTEGER             KTYPE
      REAL*8              C,           E,           T
```

2.29 etred9: Subroutine etred9 from library adas1xx

```
       FUNCTION ETRED9( KTYPE , E    , T    , B    , C )
C-----------------------------------------------------------------------
C
C  ******************** FORTRAN 77 FUNCTION: ETRED9 *********************
C
C  PURPOSE:  TO CALCULATE THE REDUCED ENERGY FOR EIGHT TYPES OF
C            TRANSITION
C
C  CALLING PROGRAM: ADAS108
C
C  FUNCTION:
C
C  INPUT:    (R*8)    E      =   TRANSITION ENERGY (Eij)
C            (R*8)    T      =   COLLIDING ELECTRON ENERGY AFTER
C                                 EXCITATION (Ej)
C            (R*8)    B      =   BURGESS SCALING PARAMETER - B
C            (R*8)    C      =   BURGESS SCALING PARAMETER - C
C            (R*8)    TL     =   Ej/Eij
C            (I*4)    KTYPE  =   TRANSITION TYPE
C                                 1 ELECTRIC DIPOLE - EXP THRESHOLD
C                                 2 NON ELECTRIC DIPOLE - EXP THRESHOLD
C                                 3 SPIN CHANGE - EXP THRESHOLD
C                                 4 OTHER - EXP THRESHOLD
C                                 5 ELECTRIC DIPOLE - POWER THRESHOLD
C                                 6 NON ELECTRIC DIPOLE - POWER THRESHOLD
C                                 7 SPIN CHANGE - POWER THRESHOLD
C                                 8 OTHER - POWER THRESHOLD
C
C  OUTPUT:   (R*8)    ETRED9 =   REDUCED ENERGY
C
C            (I*4)    IASYMC =   ASYMPTOTIC CLASSIFICATION TYPE
C            (I*4)    ITHRSC =   THRESHOLD CLASSIFICATION TYPE
C            (R*8)    E0     =   SWITCHING ENERGY
C
C
C  ROUTINES: NONE
C
C  DATE:     17/06/99 VERSION 1.1
C  AUTHOR:   HUGH SUMMERS, UNIVERSITY OF STRATHCLYDE
C
C-----------------------------------------------------------------------
      INTEGER             KTYPE
      REAL*8              B,           C,           E,           T
```

2.30 etred9inv: Subroutine etred9inv from library adas1xx

```
       FUNCTION ETRED9INV( KTYPE , EIJ   , TR    , B    , C )
C-----------------------------------------------------------------------
C
C  ******************** FORTRAN 77 FUNCTION: ETRED9INV *****************
C
C  PURPOSE:  TO CALCULATE THE ELECTRON ENERGY FROM THE REDUCED ENERGY
C            FOR EIGHT TYPES OF TRANSITION
C
C  CALLING PROGRAM: ADAS108
C
C  FUNCTION:
C
C  INPUT:    (R*8)    EIJ    =   TRANSITION ENERGY (Eij)
C            (R*8)    TR     =   REDUCED ENERGY
C            (R*8)    B      =   BURGESS SCALING PARAMETER - B
C            (R*8)    C      =   BURGESS SCALING PARAMETER - C
C            (R*8)    TL     =   Ej/Eij
C            (I*4)    KTYPE  =   TRANSITION TYPE
C                                1 ELECTRIC DIPOLE - EXP THRESHOLD
C                                2 NON ELECTRIC DIPOLE - EXP THRESHOLD
C                                3 SPIN CHANGE - EXP THRESHOLD
C                                4 OTHER - EXP THRESHOLD
C                                5 ELECTRIC DIPOLE - POWER THRESHOLD
C                                6 NON ELECTRIC DIPOLE - POWER THRESHOLD
C                                7 SPIN CHANGE - POWER THRESHOLD
C                                8 OTHER - POWER THRESHOLD
C
C  OUTPUT:   (R*8)    ETRED9INV=   ELECTRON ENERGY
C
C            (I*4)    IASYMC =   ASYMPTOTIC CLASSIFICATION TYPE
C            (I*4)    ITHRSC =   THRESHOLD CLASSIFICATION TYPE
C            (R*8)    E0     =   SWITCHING ENERGY
C
C
C  ROUTINES: NONE
C
C  DATE:     25/05/99 VERSION 1.1
C  AUTHOR:   HUGH SUMMERS, UNIVERSITY OF STRATHCLYDE
C
C-----------------------------------------------------------------------
      INTEGER             KTYPE
      REAL*8              B,           C,           EIJ,         TR
```

2.31 etredinv: Subroutine etredinv from library adas1xx

```
       FUNCTION ETREDINV( IXTYP , EIJ   , DGEX  , C )
C-----------------------------------------------------------------------
C
C  ******************** FORTRAN 77 FUNCTION: ETREDINV ******************
C
C  PURPOSE:  TO CALCULATE THE ELECTRON ENERGY FROM THE REDUCED ENERGY
C            FOR FOUR TYPES OF TRANSITION
C
C  CALLING PROGRAM:
C
C  FUNCTION:
C
C  INPUT:    (R*8)    EIJ    =   TRANSITION ENERGY (Eij)
C            (R*8)    DGEX   =   REDUCED ENERGY
C            (R*8)    C      =   ADJUSTABLE SCALING PARAMETER
C            (R*8)    TL     =   Ej/Eij
C            (I)      IXTYPE =   TRANSITION TYPE
C                                1 ELECTRIC DIPOLE
C                                2 NON ELECTRIC DIPOLE
C                                3 SPIN CHANGE
C                                4 OTHER
C
C  OUTPUT:   (R*8)    ETREDINV=   ELECTRON ENERGY
C
C
C  ROUTINES: NONE
C
C  DATE:     21/06/95 VERSION 1.1
C  AUTHOR:   DAVID.H.BROOKS (UNIV. OF STRATHCLYDE)
C
C-----------------------------------------------------------------------
      INTEGER             IXTYP
      REAL*8              C,           DGEX,        EIJ
```

2.32 five9: Subroutine five9 from library adas1xx

```
      SUBROUTINE FIVE9(A,P)
C-----------------------------------------------------------------------
C
C  ******************** FORTRAN 77 SUBROUTINE: FIVE9  ******************
C
C  PURPOSE:
C        TO CALCULATE THE BURGESS KNOT POINTS
C
C  CALLING PROGRAM:
C        LSTSQ9
C
C  INPUT:
C        (R*8) A - MATRIX CONTAINING REDUCED PARAMETERS
C  OUTPUT:
C        (R*8) P - THE BURGESS KNOT POINTS
C
C  ROUTINES:
C        MATIN1
C
C  DATE:   25/05/99 VERSION 1.1
C
C  AUTHOR: HUGH SUMMERS, UNIVERSITY OF STRATHCLYDE
C
C-----------------------------------------------------------------------
      REAL*8              A(9,9),      P(9)
```

2.33 four: Subroutine four from library adas1xx

```
      SUBROUTINE FOUR(A,B)
C-----------------------------------------------------------------------
C
C  ******************** FORTRAN 77SUBROUTINE: FOUR *********************
C
C  PURPOSE:
C        TO CALCULATE THE BURGESS KNOT POINTS
C
C  CALLING PROGRAM:
C        LSTSQ
C
C  INPUT:
C        (R*8) A - MATRIX CONTAINING REDUCED PARAMETERS
C  OUTPUT:
C        (R*8) B - THE BURGESS KNOT POINTS
C
C  ROUTINES:
C        MATIN1
C
C  DATE:   21/06/95 VERSION 1.1
C
C  AUTHOR: A.LANZAFAME (UNIV. OF STRATHCLYDE) & D.H.BROOKS
C          CONVERTED FROM BURGESS BBC BASIC
C
C-----------------------------------------------------------------------
      REAL*8             A(5,5),      B(5)
```

2.34 four9: Subroutine four9 from library adas1xx

```
      SUBROUTINE FOUR9(A,P)
C-----------------------------------------------------------------------
C
C  ******************** FORTRAN 77 SUBROUTINE: FOUR9  ******************
C
C  PURPOSE:
C        TO CALCULATE THE BURGESS KNOT POINTS
C
C  CALLING PROGRAM:
C        LSTSQ9
C
C  INPUT:
C        (R*8) A - MATRIX CONTAINING REDUCED PARAMETERS
C  OUTPUT:
C        (R*8) P - THE BURGESS KNOT POINTS
C
C  ROUTINES:
C        MATIN1
C
C  DATE:   25/05/99 VERSION 1.1
C
C  AUTHOR: HUGH SUMMERS, UNIVERSITY OF STRATHCLYDE
C
C-----------------------------------------------------------------------
      REAL*8              A(9,9),      P(9)
```

2.35 lfasym: Subroutine lfasym from library adas1xx

```
       SUBROUTINE LFASYM(X,XA,N,YA,Y,DY,C1,C2,C3,C4,FORM,IFORMS)
       IMPLICIT REAL*8(A-H,O-Z)
C-----------------------------------------------------------------------
C
C  ************  FORTRAN 77 SUBROUTINE: LFASYM ************************
C
C  PURPOSE: TO PROVIDE A SPLINE INTERPOLATE MAKING USE OF SPECIFIED
C  ASYMPTOTIC BEHAVIOUR
C
C  LARGER ARRAY DIMENSION VERSION OF NFASYM
C
C  USES LABELLED COMMON /LSPL3/
C
C  INPUT
C      X=REQUIRED X-VALUE
C      X(I)=KNOTS
C      N=NUMBER OF KNOTS
C      C1(I,J)=1ST SPINE COEFFICIENT PRECURSOR
C      C2(I,J)=2ND SPINE COEFFICIENT PRECURSOR
C      C3(I,J)=3RD SPINE COEFFICIENT PRECURSOR
C      C4(I,J)=4TH SPINE COEFFICIENT PRECURSOR
C      FORM=EXTERNAL FUNCTION SPECIFYING ASYMPTOTIC FORMS
C      IFORMS=INDEX OF REQUIRED FORM
C  OUTPUT
C      Y=RETURNED Y-VALUE
C      DY=RETURNED DERIVATIVE
C
C AUTHOR:
C
C  ********** H.P. SUMMERS, JET             7 FEB 1989   ***********
C-----------------------------------------------------------------------
C
C UNIX-IDL CONVERSION:
C
C VERSION: 1.1                          DATE: 07-10-96
C MODIFIED: WILLIAM OSBORN
C               - FIRST CONVERTED.
C
C VERSION: 1.2                          DATE: 15-05-07
C MODIFIED: Allan Whiteford
C               - Updated comments as part of subroutine
C                 documentation production.
C
C-----------------------------------------------------------------------
      INTEGER             IFORMS,      N
      REAL*8              C1(40,39),   C2(40,39),   C3(40,39)
      REAL*8              C4(40,39),   DY,          X,          XA(40)
      REAL*8              Y,           YA(40)
```

2.36 lfitsp: Subroutine lfitsp from library adas1xx

```
       SUBROUTINE LFITSP(X,XA,N,YAA,Y,DY,I0,C1,C2,C3,C4,ISW)
       IMPLICIT REAL*8(A-H,O-Z)
C-----------------------------------------------------------------------
C
C  ************  FORTRAN 77 SUBROUTINE: LFITSP ************************
C
C  PURPOSE: SUBROUTINE TO PERFORM SPLINE INTERPOLATION
C
C  LARGER ARRAY DIMENSION VERSION OF NFITSP
C
C  INPUT
C      X       = REQUIRED X-VALUE
C      XA(I)   = X-VALUES
C      N       = NUMBER OF VALUES
C      YAA(I)  = Y-VALUES (POSSIBLY STORED AS MULTIPLE SETS)
C      I0      = STARTING INDEX(-1) IN YAA ARRAY OF REQUIRED INPUT SET
C      C1(I,J) = 1ST SPLINE COEFFICIENT PRECURSOR
C      C2(I,J) = 2ND SPLINE COEFFICIENT PRECURSOR
C      C3(I,J) = 3RD SPLINE COEFFICIENT PRECURSOR
C      C4(I,J) = 4TH SPLINE COEFFICIENT PRECURSOR
C      ISW     = .LE.0  ORDINARY     SPLINE INTERPOLATION
C              = .GT.0  LOGARITHMIC  SPLINE INTERPOLATION
C  OUTPUT
C      Y       = RETURNED Y-VALUE
C      DY      = RETURNED DERIVATIVE
C
C AUTHOR:
C
C  ******** H.P.SUMMERS, JET        7 FEB 1989      **************
C
C UNIX-IDL CONVERSION:
C
C VERSION: 1.1                          DATE: 07-10-96
C MODIFIED: WILLIAM OSBORN
C               - FIRST CONVERTED.
C
C VERSION: 1.2                          DATE: 15-05-07
C MODIFIED: Allan Whiteford
C               - Updated comments as part of subroutine
C                 documentation production.
C
C-----------------------------------------------------------------------
      INTEGER             I0,          ISW,         N
      REAL*8              C1(40,39),   C2(40,39),   C3(40,39)
      REAL*8              C4(40,39),   DY,          X,           XA(40)
      REAL*8              Y,           YAA(40)
```

2.37 lgasym: Subroutine lgasym from library adas1xx

```
       SUBROUTINE LGASYM(X,DX,FORM,IFORMS,IENDS)
       IMPLICIT REAL*8(A-H,O-Z)
C-----------------------------------------------------------------------
C
C  ************  FORTRAN 77 SUBROUTINE: LGASYM ************************
C
C  PURPOSE: INITIALISES COMMON ARRAYS REQUIRED FOR SPLINING WITH
C  SMOOTH FITTING TO AN ASYMPTOTIC FORM
C
C  LARGER ARRAY DIMENSION VERSION OF NGASYM
C
C  USES LABELLED COMMON /LSPL3/
C
C  IF IENDS=1,MATCHING IS AT FIRST KNOT(GIVEN BY X)
C          =2,MATCHING IS AT  LAST KNOT(GIVEN BY X)
C  ASYMPTOTIC FORMS ARE GIVEN IN THE EXTERNAL FUNCTION FORM(I,X)
C  WHERE I=4*IFORMS-5+2*IENDS POINTS TO FIRST PART OF ASYMP. FORM
C         =4*IFORMS-4+2*IENDS POINTS TO SECOND PART OF ASYMP. FORM
C  THUS A SERIES OF ASYMPTOTIC FORMS MAY BE PRESENT IN FORM
C
C  INPUT
C      COMMON /LSPL3/ PROVIDES INPUT IN VECTOR IEND WHICH SPECIFIES
C                     CHOICE OF END CONDITION AT FIRST IEND(1) OR LAST
C                     IEND(2) KNOT OF SPLINE
C      X=X-VALUE OF END POINT
C      DX=DISPLACEMENT FROM X-VALUE FOR DERIVATIVE EVALUATION
C      FORM=EXTERNAL FUNCTION SPECIFYING ASYMPTOTIC FORMS
C      IFORMS=SELECTED FORM
C      IENDS=1,MATCHING IS AT FIRST KNOT(GIVEN BY X)
C           =2,MATCHING IS AT  LAST KNOT(GIVEN BY X)
C  OUTPUT
C      COMMON /LSPL3/ IS SET BY THIS ROUTINE
C
C AUTHOR:
C
C  ********* H.P. SUMMERS, JET           7 FEB 1989    *************
C
C-----------------------------------------------------------------------
C
C UNIX-IDL CONVERSION:
C
C VERSION: 1.1                          DATE: 07-10-96
C MODIFIED: WILLIAM OSBORN
C               - FIRST CONVERTED.
C
C VERSION: 1.2                          DATE: 15-05-07
C MODIFIED: Allan Whiteford
C               - Updated comments as part of subroutine
C                 documentation production.
C
C-----------------------------------------------------------------------
       COMMON /LSPL3/IEND(2),G(2),AB(4),PQ(12),ABRY(160)
       IF(IENDS.EQ.1.AND.IEND(1).EQ.4)GO TO 5
       IF(IENDS.EQ.2.AND.IEND(2).EQ.4)GO TO 5
    3  RETURN
    5  I=4*IFORMS-5+2*IENDS
       J=6*IENDS-5
       IC=0
       DX1=1.0D0/DX
```

```
10  PQ(J)=FORM(I,X)
    T1=FORM(I,X+DX)
    T2=FORM(I,X-DX)
    PQ(J+1)=0.5D0*DX1*(T1-T2)
    PQ(J+2)=DX1*DX1*(T1-2.0D0*PQ(J)+T2)
    IC=IC+1
    IF(IC.GT.1)RETURN
    I=I+1
    J=J+3
    GO TO 10
   END
   INTEGER             IENDS,       IFORMS
   REAL*8              DX,          X
```

2.38 lgspc: Subroutine lgspc from library adas1xx

```
       SUBROUTINE LGSPC(XA,N,C1,C2,C3,C4)
       IMPLICIT REAL*8(A-H,O-Z)
C-----------------------------------------------------------------------
C
C  ************  FORTRAN 77 SUBROUTINE: LGSPC  ************************
C
C  PURPOSE: GENERATE PRECURSORS OF SPLINE COEFFICIENTS SUITABLE FOR BOTH
C  FORWARD AND BACKWARD INTERPOLATION
C
C  LARGER ARRAY DIMENSION VERSION OF NGSPC
C
C  INPUT
C      XA(I)=SET OF KNOTS
C      N=NUMBER OF KNOTS  (N.LE.20)
C  OUTPUT
C      C1(I,J)=1ST SPLINE COEFFICIENT PRECURSOR
C      C2(I,J)=2ND SPLINE COEFFICIENT PRECURSOR
C      C3(I,J)=3RD SPLINE COEFFICIENT PRECURSOR
C      C4(I,J)=4TH SPLINE COEFFICIENT PRECURSOR
C
C AUTHOR:
C
C  ******* H.P.SUMMERS, JET         7 FEB 1989   **********
C
C UNIX-IDL CONVERSION:
C
C VERSION: 1.1                          DATE: 07-10-96
C MODIFIED: WILLIAM OSBORN
C               - FIRST CONVERTED.
C
C VERSION: 1.2                          DATE: 15-05-07
C MODIFIED: Allan Whiteford
C               - Updated comments as part of subroutine
C                 documentation production.
C
C-----------------------------------------------------------------------
      INTEGER            N
      REAL*8             C1(40,39),   C2(40,39),   C3(40,39)
      REAL*8             C4(40,39),   XA(40)
```

2.39 lsort: Subroutine lsort from library adas1xx

```
       SUBROUTINE LSORT(XA,YA,N)
       IMPLICIT REAL*8(A-H,O-Z)
C-----------------------------------------------------------------------
C
C  ************  FORTRAN 77 PROGRAM:  LSORT ************************
C
C
C  PURPOSE: SORT AN ARRAY SO THAT FIRST INPUT IS IN INCREASING ORDER
C
C  LARGER ARRAY DIMENSION VERSION OF NSORT
C
C  N.B.  INPUT VALUES ARE ALTERED BY THIS ROUTINE !!!!
C
C  INPUT
C      XA(I)=X-VALUES
C      YA(I)=Y-VALUES
C      N=NUMBER OF VALUES
C  OUTPUT
C      XA(I)=SORTED X-VALUES
C      YA(I)=SORTED Y-VALUES
C
C
C AUTHOR:
C
C  ******** H.P. SUMMERS, JET               7 FEB 1989   ***********
C
C UNIX-IDL CONVERSION:
C
C VERSION: 1.1                          DATE: 07-10-96
C MODIFIED: WILLIAM OSBORN
C               - FIRST CONVERTED.
C
C VERSION: 1.2                          DATE: 15-05-07
C MODIFIED: Allan Whiteford
C               - Updated comments as part of subroutine
C                 documentation production.
C
C-----------------------------------------------------------------------
      INTEGER             N
      REAL*8              XA(40),      YA(40)
```

2.40 lspij3: Subroutine lspij3 from library adas1xx

```
       SUBROUTINE LSPIJ3(N,H,W)
       IMPLICIT REAL*8(A-H,O-Z)
C-----------------------------------------------------------------------
C
C  ************  FORTRAN 77 SUBROUTINE: LSPIJ3 ************************
C
C  PURPOSE: CALCULATE SPLINES WITH VARIOUS END CONDITIONS.
C
C  EXTENDED ARRAY DIMENSION VERSION OF NSPIJ3
C
C  USES LABELLED COMMON  /LSPL3/
C
C  CONDITIONS AT 1ST NODE AND NTH NODE CONTROLLED BY IEND1 AND IENDN
C      IEND=1 : SPECIFIED D LOG(Y) IE. DY/Y AT NODE STORED IN APPROPRIAT5070000
C               APPROPRIATE VECTOR
C          =2 : ZERO CURVATURE
C          =3 : CONSTANT CURVATURE
C          =4 : MATCHED TO SPECIFIED FUNCTIONAL FORM IN TERMS OF
C               TWO PARAMETERS A AND B SUCH THAT
C                        FUNCT = P(1)*A+Q(1)*B
C                    1ST DERIV. = P(2)*A+Q(2)*B
C                    2ND DERIV. = P(3)*A+Q(3)*B
C               WHERE A1,B1,P1,Q1 ARE USED FOR 1ST NODE AND
C               AN,BN,PN,QN FOR NTH NODE
C
C  INPUT
C      N=NUMBER OF KNOTS
C      H(I)=INTERVALS BETWEEN KNOTS
C  OUTPUT
C      W=SPLINE MATRIX
C
C AUTHOR:
C
C  ******* H.P. SUMMERS, JET           7 FEB 1989    ***************
C
C UNIX-IDL CONVERSION:
C
C VERSION: 1.1                          DATE: 07-10-96
C MODIFIED: WILLIAM OSBORN
C               - FIRST CONVERTED.
C
C VERSION: 1.2                          DATE: 15-05-07
C MODIFIED: Allan Whiteford
C               - Updated comments as part of subroutine
C                 documentation production.
C
C-----------------------------------------------------------------------
      INTEGER             N
      REAL*8              H(40),       W(40,40)
```

2.41 lstsq: Subroutine lstsq from library adas1xx

```
      SUBROUTINE LSTSQ(IT,C,E12,GF,N,T,U,B,RMS)
C-----------------------------------------------------------------------
C  ******************** FORTRAN77 SUBROUTINE: LSTSQ ********************
C
C  PURPOSE: TO PERFORM FIVE POINT SPLINE FIT TO REDUCED OMEGAS
C
C  INPUT:
C           E12  TRANSITION ENERGY  (RYD)
C           GF   GF
C           IT   TRANSITION TYPE
C           T    ENERGY
C           U    OMEGA
C           N    NUMBER OF DATA POINTS
C           C    C PARAMETER
C
C  OUTPUT:
C           B    KNOTS VALUES
C
C  LOCAL VARIABLES/CONSTANTS:
C
C           A(,)
C           V
C           W
C           Y()
C           XX
C           YY
C
C  CALLS:
C          SPLS     - CALCULATE CUBIC SPLINE FIT COEFFICIENTS
C          ETRED    - CALCULATE REDUCED ENERGIES
C          OURED    - CALCULATE REDUCED OMEGAS
C          MATIN1   - INVERT MATRIX TO GET KNOT POINTS
C          ONE      - GET KNOT POINTS IF ONLY ONE DATA POINT
C          TWO      - GET KNOT POINTS IF ONLY TWO DATA POINTS
C          THREE    - GET KNOT POINTS IF ONLY THREE DATA POINTS
C          FOUR     - GET KNOT POINTS IF ONLY FOUR DATA POINTS
C
C  DATE: 02-07-95 VERSION 1.1
C  WRITTEN: A.LANZAFAME & D.H.BROOKS
C           UNIV.OF STRATHCLYDE,
C           CONVERTED FROM BBC BASIC
C-----------------------------------------------------------------------
      INTEGER             IT,          N
      REAL*8              B(5),        C,           E12,         GF
      REAL*8              RMS,         T(N),        U(N)
```

2.42 lstsq9: Subroutine lstsq9 from library adas1xx

```
      SUBROUTINE LSTSQ9(IT,B,C,EIJ,GF,N,T,U,P,RMS)
C-----------------------------------------------------------------------
C  ******************** FORTRAN77 SUBROUTINE: LSTSQ9 ******************
C
C  PURPOSE: TO PERFORM NINE POINT SPLINE FIT TO REDUCED OMEGAS
C
C  INPUT:
C        (R*8)   EIJ    = TRANSITION ENERGY  (RYD)
C        (R*8)   GF     = GF-VALUE
C        (I*4)   IT     = TRANSITION TYPE
C        (R*8)   T()    = ENERGIES (RYD) OF DATA POINTS
C        (R*8)   U()    = OMEGAS OF DATA POINTS
C        (I*4)   N      = NUMBER OF DATA POINTS
C        (R*8)   B      = BURGESS SCALING PARAMETER - B
C        (R*8)   C      = BURGESS SCALING PARAMETER - C
C
C  OUTPUT:
C        (R*8)   P()    = SPLINE VALUES AT BURGESS/SUMMERS KNOTS
C
C  LOCAL VARIABLES/CONSTANTS:
C
C        (R*8)   A(,)   = NORMAL EQUATION COEFFICIENT MATRIX
C        (R*8)   V      = GENERAL VARIABLE
C        (R*8)   W      = REAL CONSTANT (=3)
C        (R*8)   Y()    = TEMPORARY KNOTS
C        (R*8)   XX     = GENERAL VARIABLE
C        (R*8)   YY     = GENERAL VARIABLE
C
C  ROUTINES:
C          SPLS9    - CALCULATE CUBIC SPLINE FIT COEFFICIENTS
C          ETRED9   - CALCULATE REDUCED ENERGIES
C          OURED9   - CALCULATE REDUCED OMEGAS
C          MATIN1   - INVERT MATRIX TO GET KNOT POINTS
C          ONE9     - GET KNOT POINTS IF ONLY ONE DATA POINT
C          TWO9     - GET KNOT POINTS IF ONLY TWO DATA POINTS
C          THREE9   - GET KNOT POINTS IF ONLY THREE DATA POINTS
C          FOUR9    - GET KNOT POINTS IF ONLY FOUR DATA POINTS
C          FIVE9    - GET KNOT POINTS IF ONLY FIVE DATA POINT
C          SIX9     - GET KNOT POINTS IF ONLY SIX DATA POINTS
C          SEVEN9   - GET KNOT POINTS IF ONLY SEVEN DATA POINTS
C          EIGHT9   - GET KNOT POINTS IF ONLY EIGHT DATA POINTS
C
C  DATE:     25-05-99 VERSION 1.1
C  AUTHOR:   HUGH SUMMERS, UNIVERSITY OF STRATHCLYDE
C      FIRST RELEASE
C
C-----------------------------------------------------------------------
      INTEGER             IT,          N
      REAL*8              B,           C,           EIJ,         GF
      REAL*8              P(9),        RMS,         T(N),        U(N)
```

2.43 maxw9: Subroutine maxw9 from library adas1xx

```
      SUBROUTINE MAXW9( IT    , B     , C    , E    ,
     &                  P     , NT    , T     , U
     &                )
C-----------------------------------------------------------------------
C  ******************  FORTRAN77 SUBROUTINE: MAXW9   ******************
C
C  PURPOSE: GAUSS-LAGUERRE QUADRATURE FROM BURGESS' PROGRAM
C           OMEUPS
C
C  INPUT:
C       (I*4)  IT       = TRANSITION TYPE
C       (R*8)  B        = BURGESS/SUMMERS SCALING PARAMETER - B
C       (R*8)  C        = BURGESS/SUMMERS SCALING PARAMETER - C
C       (R*8)  E        = EXCITATION ENERGY
C       (R*8)  P()      = KNOTS
C       (I*4)  NT       = NUMBER OF TEMPERATURE POINTS
C       (R*8)  T()      = TEMPERATURES (K)
C
C  OUTPUT:
C       (R*8)  U()      = UPSILONS
C
C  ROUTINES:
C    OMEUPS9 - TO CALCULATE THE MAXWELL AVERAGED UPSILONS
C    UPS9    - TO CALCULATE THE MAXWELL AVERAGED UPSILONS FOR ALL
C              TRANSITIONS EXCEPT 'SPIN CHANGE'
C
C  DATE:     24-06-99 VERSION: 1.1
C  AUTHOR:   HUGH SUMMERS, UNIVERSITY OF STRATHCLYDE
C - FIRST RELEASE
C
C-----------------------------------------------------------------------
      INTEGER             IT,          NT
      REAL*8              B,           C,           E,           P(9)
      REAL*8              T(NT),       U(NT)
```

2.44 maxwell: Subroutine maxwell from library adas1xx

```
      SUBROUTINE MAXWELL(IT,C,E,P,NT,T,U)
C-----------------------------------------------------------------------
C  ******************  FORTRAN77 SUBROUTINE: MAXWELL ******************
C
C  PURPOSE: GAUSS-LAGUERRE QUADRATURE FROM BURGESS' PROGRAM
C           OMEUPS
C
C  INPUT:
C    IT  TRANSITION TYPE
C    C   OMEGA SCALING PARAMETER
C    E   EXCITATION ENERGY
C    P   KNOTS
C    NT  NUMBER OF TEMPERATURE POINTS
C    T   TEMPERATURE
C
C  OUTPUT:
C    U   UPSILON
C
C  ROUTINES:
C    OMEUPS - TO CALCULATE THE MAXWELL AVERAGED UPSILONS
C    UPS    - TO CALCULATE THE MAXWELL AVERAGED UPSILONS FOR ALL
C             TRANSITIONS EXCEPT 'SPIN CHANGE'
C
C  WRITTEN:
C        A.LANZAFAME & D.H.BROOKS, UNIV.OF STRATHCLYDE, 02-JULY-1995
C        CONVERTED FROM BURGESS BBC BASIC
C  DATE: 02-07-95 VERSION: 1.1
C  MODIFIED: A.LANZAFAME & D.H.BROOKS
C - FIRST RELEASE
C
C-----------------------------------------------------------------------
      INTEGER             IT,          NT
      REAL*8              C,           E,           P(5),        T(NT)
      REAL*8              U(NT)
```

2.45 nfasym: Subroutine nfasym from library adas1xx

```
       SUBROUTINE NFASYM(X,XA,N,YA,Y,DY,C1,C2,C3,C4,FORM,IFORMS)
       IMPLICIT REAL*8(A-H,O-Z)
C-----------------------------------------------------------------------
C  PURPOSE: PROVIDE A SPLINE INTERPOLATE MAKING USE OF SPECIFIED
C  ASYMPTOTIC BEHAVIOUR
C
C  USES LABELLED COMMON /SPL3/
C
C  INPUT
C      X=REQUIRED X-VALUE
C      X(I)=KNOTS
C      N=NUMBER OF KNOTS
C      C1(I,J)=1ST SPINE COEFFICIENT PRECURSOR
C      C2(I,J)=2ND SPINE COEFFICIENT PRECURSOR
C      C3(I,J)=3RD SPINE COEFFICIENT PRECURSOR
C      C4(I,J)=4TH SPINE COEFFICIENT PRECURSOR
C      FORM=EXTERNAL FUNCTION SPECIFYING ASYMPTOTIC FORMS
C      IFORMS=INDEX OF REQUIRED FORM
C  OUTPUT
C      Y=RETURNED Y-VALUE
C      DY=RETURNED DERIVATIVE
C
C
C  ***********                                          ***********
C-----------------------------------------------------------------------
C IDL-UNIX CONVERSION:
C
C  VERSION: 1.1                                    DATE: 01/11/96
C  MODIFIED:  WILLIAM OSBORN
C           - FIRST WRITTEN. NO CHANGES.
C
C  VERSION: 1.2                                    DATE: 15/05/07
C  MODIFIED:  Allan Whiteford
C           - Updated comments as part of subroutine
C             documentation production.
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      INTEGER             IFORMS,      N
      REAL*8              C1(10,9),    C2(10,9),    C3(10,9)
      REAL*8              C4(10,9),    DY,          X,           XA(10)
      REAL*8              Y,           YA(10)
```

2.46 nfitsp: Subroutine nfitsp from library adas1xx

```
       SUBROUTINE NFITSP(X,XA,N,YAA,Y,DY,I0,C1,C2,C3,C4,ISW)
       IMPLICIT REAL*8(A-H,O-Z)
C-----------------------------------------------------------------------
C  PURPOSE: PERFORM SPLINE INTERPOLATION
C
C  INPUT
C      X       = REQUIRED X-VALUE
C      XA(I)   = X-VALUES
C      N       = NUMBER OF VALUES
C      YAA(I)  = Y-VALUES (POSSIBLY STORED AS MULTIPLE SETS)
C      I0      = STARTING INDEX(-1) IN YAA ARRAY OF REQUIRED INPUT SET
C      C1(I,J) = 1ST SPLINE COEFFICIENT PRECURSOR
C      C2(I,J) = 2ND SPLINE COEFFICIENT PRECURSOR
C      C3(I,J) = 3RD SPLINE COEFFICIENT PRECURSOR
C      C4(I,J) = 4TH SPLINE COEFFICIENT PRECURSOR
C      ISW     = .LE.0  ORDINARY     SPLINE INTERPOLATION
C              = .GT.0  LOGARITHMIC  SPLINE INTERPOLATION
C  OUTPUT
C      Y       = RETURNED Y-VALUE
C      DY      = RETURNED DERIVATIVE
C
C
C  *********                                              **************
C-----------------------------------------------------------------------
C IDL-UNIX CONVERSION:
C
C  VERSION: 1.1                                          DATE: 01/10/96
C  MODIFIED:  WILLIAM OSBORN
C           - FIRST WRITTEN. NO CHANGES.
C
C  VERSION: 1.2                                          DATE: 15/05/07
C  MODIFIED:  Allan Whiteford
C           - Updated comments as part of subroutine
C             documentation production.
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      INTEGER             I0,           ISW,          N
      REAL*8              C1(10,9),     C2(10,9),     C3(10,9)
      REAL*8              C4(10,9),     DY,           X,           XA(10)
      REAL*8              Y,            YAA(10)
```

2.47 ngasym: Subroutine ngasym from library adas1xx

```
       SUBROUTINE NGASYM(X,DX,FORM,IFORMS,IENDS)
       IMPLICIT REAL*8(A-H,O-Z)
C-----------------------------------------------------------------------
C  PURPOSE: INITIALISES COMMON ARRAYS REQUIRED FOR SPLINING WITH
C  SMOOTH FITTING TO AN ASYMPTOTIC FORM
C
C
C
C  USES LABELLED COMMON /SPL3/
C
C  IF IENDS=1,MATCHING IS AT FIRST KNOT(GIVEN BY X)
C          =2,MATCHING IS AT  LAST KNOT(GIVEN BY X)
C  ASYMPTOTIC FORMS ARE GIVEN IN THE EXTERNAL FUNCTION FORM(I,X)
C  WHERE I=4*IFORMS-5+2*IENDS POINTS TO FIRST PART OF ASYMP. FORM
C         =4*IFORMS-4+2*IENDS POINTS TO SECOND PART OF ASYMP. FORM
C  THUS A SERIES OF ASYMPTOTIC FORMS MAY BE PRESENT IN FORM
C
C  INPUT
C      COMMON /SPL3/ PROVIDES INPUT IN VECTOR IEND WHICH SPECIFIES
C                    CHOICE OF END CONDITION AT FIRST IEND(1) OR LAST
C                    IEND(2) KNOT OF SPLINE
C      X=X-VALUE OF END POINT
C      DX=DISPLACEMENT FROM X-VALUE FOR DERIVATIVE EVALUATION
C      FORM=EXTERNAL FUNCTION SPECIFYING ASYMPTOTIC FORMS
C      IFORMS=SELECTED FORM
C      IENDS=1,MATCHING IS AT FIRST KNOT(GIVEN BY X)
C           =2,MATCHING IS AT  LAST KNOT(GIVEN BY X)
C  OUTPUT
C      COMMON /SPL3/ IS SET BY THIS ROUTINE
C
C
C  *************                                          *************
C-----------------------------------------------------------------------
C IDL-UNIX CONVERSION:
C
C  VERSION: 1.1                                          DATE: 01/10/96
C  MODIFIED:  WILLIAM OSBORN
C           - FIRST WRITTEN. NO CHANGES.
C
C  VERSION: 1.2                                          DATE: 15/05/07
C  MODIFIED:  Allan Whiteford
C           - Updated comments as part of subroutine
C             documentation production.
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
       COMMON /SPL3/IEND(2),G(2),AB(4),PQ(12),ABRY(40)
       IF(IENDS.EQ.1.AND.IEND(1).EQ.4)GO TO 5
       IF(IENDS.EQ.2.AND.IEND(2).EQ.4)GO TO 5
    3  RETURN
    5  I=4*IFORMS-5+2*IENDS
       J=6*IENDS-5
       IC=0
       DX1=1.0D0/DX
   10  PQ(J)=FORM(I,X)
       T1=FORM(I,X+DX)
       T2=FORM(I,X-DX)
       PQ(J+1)=0.5D0*DX1*(T1-T2)
       PQ(J+2)=DX1*DX1*(T1-2.0D0*PQ(J)+T2)
```

```
 IC=IC+1
 IF(IC.GT.1)RETURN
 I=I+1
 J=J+3
 GO TO 10
END
INTEGER             IENDS,       IFORMS
REAL*8              DX,          X
```

2.48 ngspc: Subroutine ngspc from library adas1xx

```
       SUBROUTINE NGSPC(XA,N,C1,C2,C3,C4)
       IMPLICIT REAL*8(A-H,O-Z)
C-----------------------------------------------------------------------
C  PURPOSE: GENERATE PRECURSORS OF SPLINE COEFFICIENTS SUITABLE
C  FOR BOTH FORWARD AND BACKWARD INTERPOLATION
C
C  INPUT
C      XA(I)=SET OF KNOTS
C      N=NUMBER OF KNOTS  (N.LE.20)
C  OUTPUT
C      C1(I,J)=1ST SPLINE COEFFICIENT PRECURSOR
C      C2(I,J)=2ND SPLINE COEFFICIENT PRECURSOR
C      C3(I,J)=3RD SPLINE COEFFICIENT PRECURSOR
C      C4(I,J)=4TH SPLINE COEFFICIENT PRECURSOR
C
C
C
C-----------------------------------------------------------------------
C IDL-UNIX CONVERSION:
C
C  VERSION: 1.1                                        DATE: 01/10/96
C  MODIFIED:  WILLIAM OSBORN
C           - FIRST WRITTEN. NO CHANGES.
C
C  VERSION: 1.2                                        DATE: 15/05/07
C  MODIFIED:  Allan Whiteford
C           - Updated comments as part of subroutine
C             documentation production.
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      INTEGER            N
      REAL*8             C1(10,9),    C2(10,9),    C3(10,9)
      REAL*8             C4(10,9),    XA(10)
```

2.49 nspij3: Subroutine nspij3 from library adas1xx

```
       SUBROUTINE NSPIJ3(N,H,W)
       IMPLICIT REAL*8(A-H,O-Z)
C-----------------------------------------------------------------------
C  PURPOSE: CALCULATE SPLINES WITH VARIOUS END CONDITIONS.
C
C  USES LABELLED COMMON  /SPL3/
C
C  CONDITIONS AT 1ST NODE AND NTH NODE CONTROLLED BY IEND1 AND IENDN
C      IEND=1 : SPECIFIED D LOG(Y) IE. DY/Y AT NODE STORED IN APPROPRIAT
C               APPROPRIATE VECTOR
C          =2 : ZERO CURVATURE
C          =3 : CONSTANT CURVATURE
C          =4 : MATCHED TO SPECIFIED FUNCTIONAL FORM IN TERMS OF
C               TWO PARAMETERS A AND B SUCH THAT
C                        FUNCT = P(1)*A+Q(1)*B
C                    1ST DERIV. = P(2)*A+Q(2)*B
C                    2ND DERIV. = P(3)*A+Q(3)*B
C               WHERE A1,B1,P1,Q1 ARE USED FOR 1ST NODE AND
C               AN,BN,PN,QN FOR NTH NODE
C
C  INPUT
C      N=NUMBER OF KNOTS
C      H(I)=INTERVALS BETWEEN KNOTS
C  OUTPUT
C      W=SPLINE MATRIX
C
C  *********                                              **************
C-----------------------------------------------------------------------
C IDL-UNIX CONVERSION:
C
C  VERSION: 1.1                                          DATE: 01/10/96
C  MODIFIED:  WILLIAM OSBORN
C           - FIRST WRITTEN. NO CHANGES.
C
C  VERSION: 1.2                                          DATE: 15/05/07
C  MODIFIED:  Allan Whiteford
C           - Updated comments as part of subroutine
C             documentation production.
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      INTEGER            N
      REAL*8             H(10),       W(10,10)
```

2.50 omeups: Subroutine omeups from library adas1xx

```
      FUNCTION OMEUPS(IT,E,C,P,X)
C-----------------------------------------------------------------------
C
C  ******************** FORTRAN77 SUBROUTINE:OMEUPS*********************
C
C  PURPOSE:
C        TO CALCULATE UPSILONS
C
C  CALLING PROGRAM:
C        MAXWELL
C
C  INPUT:
C        (I)     IT - TRANSITION TYPE
C        (R*8)    E - EXCITATION ENRGY
C        (R*8)    C - SCALABLE PARAMETER
C        (R*8)    P - KNOT POINTS
C        (R*8)    X - COLLIDING ELECTRON ENERGY AFTER
C                     EXCITATION
C  OUTPUT:
C        (R*8) OMEUPS - THE UPSILONS
C
C  ROUTINES:
C
C  DATE:    21/06/95 VERSION 1.1
C  AUTHOR:  A.LANZAFAME (UNIV. OF STRATHCLYDE) & D.H.BROOKS
C           CONVERTED FROM BURGESS BBC BASIC
C
C-----------------------------------------------------------------------
      INTEGER             IT
      REAL*8              C,           E,           P(5),        X
```

2.51 omeups9: Subroutine omeups9 from library adas1xx

```
      FUNCTION OMEUPS9( IT   , E    , B    , C    , P    , X )
C-----------------------------------------------------------------------
C
C  ******************** FORTRAN77 SUBROUTINE:OMEUPS9*******************
C
C  PURPOSE:
C        TO CALCULATE UPSILONS
C
C  CALLING PROGRAM:
C        MAXW9
C
C  INPUT:
C        (I*4)  IT      = TRANSITION TYPE
C        (R*8)  E       = EXCITATION ENERGY (RYD)
C        (R*8)  B       = BURGESS SCALABLE PARAMETER - B
C        (R*8)  C       = BURGESS SCALABLE PARAMETER - C
C        (R*8)  P()     = SPLINE VALUES AT KNOTS
C        (R*8)  X       = COLLIDING ELECTRON ENERGY AFTER EXCITATION
C  OUTPUT:
C        (R*8)  OMEUPS9 = UPSILON
C
C  ROUTINES:
C         SP9    - BURGESS SPLINE FIT
C         OMUP9  -
C
C  DATE:    21/06/95 VERSION 1.1
C  AUTHOR:  A.LANZAFAME (UNIV. OF STRATHCLYDE) & D.H.BROOKS
C           CONVERTED FROM BURGESS BBC BASIC
C
C-----------------------------------------------------------------------
      INTEGER             IT
      REAL*8              B,           C,           E,           P(9)
      REAL*8              X
```

2.52 omup: Subroutine omup from library adas1xx

```
      FUNCTION OMUP(IT,E,T,U,C)
C-----------------------------------------------------------------------
C
C  ******************** FORTRAN77 SUBROUTINE: OMUP *********************
C
C  PURPOSE:
C        TO CALCULATE UPSILONS FOR DIFFERENT TRANSITIONS
C
C  CALLING PROGRAM:
C        OMEUPS
C
C  INPUT:
C        (I)     IT - TRANSITION TYPE
C        (R*8)    E - EXCITATION ENRGY
C        (R*8)    T - SCALED ENERGY VALUE OF QUADRATURE FIXED POINTS
C        (R*8)    U - SPLINE FIT TO THE KNOT POINTS AND
C                     REDUCED ENERGIES
C        (R*8)    C - SCALABLE PARAMETER
C  OUTPUT:
C        (R*8) OMUP - THE UPSILONS
C
C
C  DATE:   21/06/95 VERSION 1.1
C  AUTHOR: A.LANZAFAME (UNIV. OF STRATHCLYDE) & D.H.BROOKS
C          CONVERTED FROM BURGESS BBC BASIC
C
C  DATE:   25/11/96 VERSION 1.2
C  AUTHOR: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C          CORRECTED IT.EQ.3 OPTION
C
C  DATE:   13/05/99 VERSION 1.3
C  AUTHOR: HUGH SUMMERS, UNIVERSITY OF STRATHCLYDE
C          PREVENTED CHANGE OF INPUT SUBROUTINE PARAMETER T ON RETURN
C
C-----------------------------------------------------------------------
      INTEGER             IT
      REAL*8              C,           E,           T,           U
```

2.53 omup9: Subroutine omup9 from library adas1xx

```
      FUNCTION OMUP9( KTYPE , E    , T    , U    , B     , C )
C-----------------------------------------------------------------------
C
C  ******************** FORTRAN77 SUBROUTINE: OMUP9 *******************
C
C  PURPOSE:
C        TO CALCULATE UPSILONS FOR DIFFERENT TRANSITIONS
C
C  CALLING PROGRAM:
C        OMEUPS
C
C  INPUT:
C        (I*4)   KTYPE  = TRANSITION TYPE
C        (R*8)   E      = EXCITATION ENERGY (RYD)
C        (R*8)   T      = SCALED ENERGY VALUE OF QUADRATURE FIXED POINTS
C        (R*8)   U      = SPLINE FIT TO THE KNOT POINTS AT RED. ENERGIES
C        (R*8)   B      = BURGESS SCALABLE PARAMETER - B
C        (R*8)   C      = BURGESS SCALABLE PARAMETER - C
C  OUTPUT:
C        (R*8)  OMUP9   = UPSILON
C
C
C  DATE:   25/05/99 VERSION 1.1
C  AUTHOR: HUGH SUMMERS, UNIVERSITY OF STRATHCLYDE
C
C-----------------------------------------------------------------------
      INTEGER             KTYPE
      REAL*8              B,           C,           E,           T
      REAL*8              U
```

2.54 one: Subroutine one from library adas1xx

```
      SUBROUTINE ONE(A,B)
C-----------------------------------------------------------------------
C
C  ******************** FORTRAN 77SUBROUTINE: ONE  *********************
C
C  PURPOSE:
C        TO CALCULATE THE BURGESS KNOT POINTS
C
C  CALLING PROGRAM:
C        LSTSQ
C
C  INPUT:
C        (R*8) A - MATRIX CONTAINING REDUCED PARAMETERS
C  OUTPUT:
C        (R*8) B - THE BURGESS KNOT POINTS
C
C  ROUTINES:
C        MATIN1
C
C  DATE:   21/06/95 VERSION 1.1
C
C  AUTHOR: A.LANZAFAME (UNIV. OF STRATHCLYDE) & D.H.BROOKS
C          CONVERTED FROM BURGESS BBC BASIC
C
C-----------------------------------------------------------------------
      REAL*8             A(5,5),      B(5)
```

2.55 one9: Subroutine one9 from library adas1xx

```
      SUBROUTINE ONE9(A,P)
C-----------------------------------------------------------------------
C
C  ******************** FORTRAN 77 SUBROUTINE: ONE9  ********************
C
C  PURPOSE:
C        TO CALCULATE THE NINE BURGESS KNOT POINTS
C
C  CALLING PROGRAM:
C        LSTSQ9
C
C  INPUT:
C        (R*8) A - MATRIX CONTAINING REDUCED PARAMETERS
C  OUTPUT:
C        (R*8) P - THE NINE BURGESS KNOT POINTS
C
C  ROUTINES:
C        NONE
C
C  DATE:    25/05/99 VERSION 1.1
C
C  AUTHOR:  HUGH SUMMERS, UNIVERSITY OF STRATHCLYDE
C           FIRST RELEASE
C
C-----------------------------------------------------------------------
      REAL*8              A(9,9),      P(9)
```

2.56 oured: Subroutine oured from library adas1xx

```
       FUNCTION OURED( KTYPE, EIJ  , EJ   , OMUP , C    )
C-----------------------------------------------------------------------
C
C  ******************** FORTRAN 77 FUNCTION: OURED *********************
C
C  PURPOSE:  TO CALCULATE THE REDUCED COLLISION STRENGTH AS A FUNCTION
C            OF Ej/Eij FOR FOUR TYPES OF TRANSITION
C
C  CALLING PROGRAM: VARIOUS ADAS101 ROUTINES
C
C  FUNCTION:
C
C  INPUT:    (R*8)    EIJ    =   TRANSITION ENERGY (Eij)
C            (R*8)    EJ     =   COLLIDING ELECTRON ENERGY AFTER
C                                  EXCITATION (Ej)
C            (R*8)    C      =   ADJUSTABLE SCALING PARAMETER
C            (R*8)    ETR     =  Ej/Eij
C            (R*8)    OMUP   =   COLLISION STRENGTH AS A FUNCTION
C                                  OF ETR
C            (I)      KTYPE  =   TRANSITION TYPE
C                                  1 ELECTRIC DIPOLE
C                                  2 NON ELECTRIC DIPOLE
C                                  3 SPIN CHANGE
C                                  4 OTHER
C
C  OUTPUT:   (R*8)    OURED  =   REDUCED COLLISION STRENGTH
C
C
C  ROUTINES: NONE
C
C  DATE:     21/06/95 VERSION 1.1
C  AUTHOR:   A.LANZAFAME (UNIV. OF STRATHCLYDE) & D.H.BROOKS
C
C-----------------------------------------------------------------------
      INTEGER             KTYPE
      REAL*8              C,           EIJ,         EJ,          OMUP
```

2.57 oured9: Subroutine oured9 from library adas1xx

```
       FUNCTION OURED9( KTYPE, EIJ  , EJ   , OMUP , B    , C    )
C-----------------------------------------------------------------------
C
C  ******************** FORTRAN 77 FUNCTION: OURED9 *******************
C
C  PURPOSE:  TO CALCULATE THE REDUCED COLLISION STRENGTH AS A FUNCTION
C            OF Ej/Eij FOR EIGHT TYPES OF TRANSITION
C
C  CALLING PROGRAM: VARIOUS ADAS108 ROUTINES
C
C  FUNCTION:
C
C  INPUT:    (R*8)    EIJ    =   TRANSITION ENERGY (RYD)
C            (R*8)    EJ     =   COLLIDING ELECTRON ENERGY AFTER
C                                 EXCITATION (RYD)
C            (R*8)    C      =   BURGESS SCALING PARAMETER -C
C            (R*8)    C      =   BURGESS SCALING PARAMETER -C
C            (R*8)    ETR    =   Ej/Eij
C            (R*8)    OMUP   =   COLLISION STRENGTH AS A FUNCTION
C                                 OF ETR
C            (I*4)    KTYPE  =   TRANSITION TYPE
C                                 1 ELECTRIC DIPOLE - EXP THRESHOLD
C                                 2 NON ELECTRIC DIPOLE - EXP THRESHOLD
C                                 3 SPIN CHANGE - EXP THRESHOLD
C                                 4 OTHER - EXP THRESHOLD
C                                 5 ELECTRIC DIPOLE - POWER THRESHOLD
C                                 6 NON ELECTRIC DIPOLE - POWER THRESHOLD
C                                 7 SPIN CHANGE - POWER THRESHOLD
C                                 8 OTHER - POWER THRESHOLD
C
C  OUTPUT:   (R*8)    OURED9 =   REDUCED COLLISION STRENGTH
C
C            (I*4)    IASYMC =   ASYMPTOTIC CLASSIFICATION TYPE
C            (I*4)    ITHRSC =   THRESHOLD CLASSIFICATION TYPE
C            (R*8)    E0     =   SWITCHING ENERGY
C
C
C  ROUTINES: NONE
C
C  DATE:     17/06/99 VERSION 1.1
C  AUTHOR:   HUGH SUMMERS, UNIVERSITY OF STRATHCLYDE
C
C-----------------------------------------------------------------------
      INTEGER             KTYPE
      REAL*8              B,           C,           EIJ,         EJ
      REAL*8              OMUP
```

2.58 oured9inv: Subroutine oured9inv from library adas1xx

```
       FUNCTION OURED9INV( KTYPE , EIJ   , EJ  , OURED  , B    , C )
C-----------------------------------------------------------------------
C
C  ******************** FORTRAN 77 FUNCTION: OURED9INV *****************
C
C  PURPOSE:  TO CALCULATE THE COLLISION STRENGTH FROM THE REDUCED
C            COLLISION STRENGTH FOR EIGHT TYPES OF TRANSITION
C
C  CALLING PROGRAM:
C
C  FUNCTION:
C
C  INPUT:    (R*8)    EIJ    =   TRANSITION ENERGY (RYD)
C            (R*8)    EJ     =   COLLIDING ELECTRON ENERGY AFTER
C                                 EXCITATION (RYD)
C            (R*8)    B      =   BURGESS SCALING PARAMETER - B
C            (R*8)    C      =   BURGESS SCALING PARAMETER - C
C            (R*8)    TL     =   Ej/Eij
C            (I)      KTYPE  =   TRANSITION TYPE
C                                 1 ELECTRIC DIPOLE
C                                 2 NON ELECTRIC DIPOLE
C                                 3 SPIN CHANGE
C                                 4 OTHER
C            (R*8)    OURED   =   REDUCED COLLISION STRENGTH
C
C  OUTPUT:   (R*8)    OURED9INV =   UPSILON
C
C            (I*4)    IASYMC =   ASYMPTOTIC CLASSIFICATION TYPE
C            (I*4)    ITHRSC =   THRESHOLD CLASSIFICATION TYPE
C            (R*8)    E0     =   SWITCHING ENERGY
C
C
C  ROUTINES: NONE
C
C  DATE:     25/05/99 VERSION 1.1
C  AUTHOR:   HUGH SUMMERS, UNIVERSITY OF STRATHCLYDE
C
C
C-----------------------------------------------------------------------
      INTEGER             KTYPE
      REAL*8              B,           C,           EIJ,         EJ
      REAL*8              OURED
```

2.59 ouredinv: Subroutine ouredinv from library adas1xx

```
       FUNCTION OUREDINV( IXTYP , EIJ   , EJ  , DGEY  , C )
C-----------------------------------------------------------------------
C
C  ******************** FORTRAN 77 FUNCTION: ETREDINV ******************
C
C  PURPOSE:  TO CALCULATE THE COLLISION STRENGTH FROM THE REDUCED
C            COLLISION STRENGTH FOR FOUR TYPES OF TRANSITION
C
C  CALLING PROGRAM:
C
C  FUNCTION:
C
C  INPUT:    (R*8)    EIJ    =   TRANSITION ENERGY (Eij)
C            (R*8)    EJ     =   COLLIDING ELECTRON ENERGY AFTER
C                                 EXCITATION (Ej)
C            (R*8)    C      =   ADJUSTABLE SCALING PARAMETER
C            (R*8)    ETR    =   Ej/Eij
C            (I)      IXTYPE =   TRANSITION TYPE
C                                 1 ELECTRIC DIPOLE
C                                 2 NON ELECTRIC DIPOLE
C                                 3 SPIN CHANGE
C                                 4 OTHER
C            (R*8)    DGEY   =   REDUCED COLLISION STRENGTH
C
C  OUTPUT:   (R*8)    OUREDINV=   ELECTRON ENERGY
C
C
C  ROUTINES: NONE
C
C  DATE:     22/06/95 VERSION 1.1
C  AUTHOR:   DAVID.H.BROOKS (UNIV. OF STRATHCLYDE)
C
C
C-----------------------------------------------------------------------
      INTEGER             IXTYP
      REAL*8              C,           DGEY,        EIJ,         EJ
```

2.60 qbchid: Subroutine qbchid from library adas1xx

```
       FUNCTION QBCHID(Z,XI,ZETA,ENER)
       IMPLICIT REAL*8(A-H,O-Z)
C-----------------------------------------------------------------------
C
C  ******************** FORTRAN77 SUBROUTINE QBCHID ******************
C
C  PURPOSE:
C   EVALUATES A SHELL CONTRIBUTION TO THE IONISATION RATE COEFFICIENT
C   IN THE BURGESS-CHIDICHIMO APPROXIMATION  MNRAS(1983)203,1269.
C   EXCLUDING THE THRESHOLD CORRECTION FACTOR
C
C  CALLING PROGRAM:
C            SPFMAN11.FOR
C
C  INPUT:
C      Z=TARGET ION CHARGE NUMBER
C      XI=EFFECTIVE IONISATION POTENTIAL FOR SHELL (RYD)
C      ZETA=EFFECTIVE NUMBER OF EQUIVALENT ELECTRONS IN SHELL
C      ENER=ELECTRON ENERGY (RYD)
C
C  OUTPUT:
C      QBCHID=IONISATION CROSS-SECTION (PI*A0**2)
C
C  AUTHOR:
C      H.P SUMMERS, JET            1 JULY 1987
C
C UNIX-IDL CONVERSION:
C
C  VERSION 1.1                                          DATE: 29-08-96
C  MODIFIED: WILLIAM OSBORN, TESSELLA SUPPORT SERVICES PLC.
C - NO FUNCTIONAL CHANGES TO IBM VERSION.
C
C-----------------------------------------------------------------------
       QBCHID=0.0D0
       IF(ENER.LE.XI)RETURN
       C=2.3D0
       BETA=0.25D0*(DSQRT((100.0D0*Z+91.0)/(4.0D0*Z+3.0D0))-5.0D0)
       T1=ZETA*DLOG(ENER/XI)/(XI*ENER)
       W=(DLOG(ENER/XI))**(BETA*XI/ENER)
C  OMIT THE W THRESHOLD ADJUSTMENT
C      QBCHID=C*T1*W
       QBCHID=C*T1
   10  RETURN
      END
      REAL*8              ENER,        XI,          Z,           ZETA
```

2.61 seven9: Subroutine seven9 from library adas1xx

```
      SUBROUTINE SEVEN9(A,P)
C-----------------------------------------------------------------------
C
C  ******************** FORTRAN 77 SUBROUTINE: SEVEN9 ******************
C
C  PURPOSE:
C        TO CALCULATE THE BURGESS KNOT POINTS
C
C  CALLING PROGRAM:
C        LSTSQ9
C
C  INPUT:
C        (R*8) A - MATRIX CONTAINING REDUCED PARAMETERS
C  OUTPUT:
C        (R*8) P - THE BURGESS KNOT POINTS
C
C  ROUTINES:
C        MATIN1
C
C  DATE:   25/05/99 VERSION 1.1
C
C  AUTHOR: HUGH SUMMERS, UNIVERSITY OF STRATHCLYDE
C
C-----------------------------------------------------------------------
      REAL*8              A(9,9),      P(9)
```

2.62 six9: Subroutine six9 from library adas1xx

```
      SUBROUTINE SIX9(A,P)
C-----------------------------------------------------------------------
C
C  ******************** FORTRAN 77 SUBROUTINE: SIX9  ********************
C
C  PURPOSE:
C        TO CALCULATE THE BURGESS KNOT POINTS
C
C  CALLING PROGRAM:
C        LSTSQ9
C
C  INPUT:
C        (R*8) A - MATRIX CONTAINING REDUCED PARAMETERS
C  OUTPUT:
C        (R*8) P - THE BURGESS KNOT POINTS
C
C  ROUTINES:
C        MATIN1
C
C  DATE:   25/05/99 VERSION 1.1
C
C  AUTHOR: HUGH SUMMERS, UNIVERSITY OF STRATHCLYDE
C
C-----------------------------------------------------------------------
      REAL*8              A(9,9),      P(9)
```

2.63 sp5: Subroutine sp5 from library adas1xx

```
      FUNCTION SP5(P,X)
C-----------------------------------------------------------------------
C
C  ******************** FORTRAN 77 FUNCTION: SP5 ***********************
C
C  PURPOSE:  TO CALCULATE A SPLINE THROUGH THE FIVE KNOT POINTS
C
C  CALLING PROGRAM:
C        OMEUPS
C
C
C  INPUT:
C        (R*8) P    - THE FIVE KNOT POINTS
C        (R*8) X    - THE POINT AT WHICH TO EVALUATE THE SPLINE
C  OUTPUT:
C        (R*8) SP5  - THE EVALUATED SPLINE
C
C  ROUTINES: NONE
C
C  DATE:     21/06/95 VERSION 1.1
C  AUTHOR:   A.LANZAFAME (UNIV. OF STRATHCLYDE) & D.H.BROOKS
C
C  DATE:     13/05/99 VERSION 1.2
C  AUTHOR:   HUGH SUMMERS, UNIVERSITY OF STRATHCLYDE
C            CORRECTED SERIOUS ERROR IN IF JUMPS AND MODIFICATION OF
C            INPUT VARIABLE X ON RETURN.  MADE CONSTANTS DOUBLE PREC.
C
C
C-----------------------------------------------------------------------
      REAL*8              P(5),        X
```

2.64 sp9: Subroutine sp9 from library adas1xx

```
      FUNCTION SP9(P,X)
C-----------------------------------------------------------------------
C
C  ******************** FORTRAN 77 FUNCTION: SP9 ***********************
C
C  PURPOSE:  TO CALCULATE A SPLINE THROUGH THE NINE KNOT POINTS
C
C  CALLING PROGRAM:
C        OMEUPS
C
C
C  INPUT:
C        (R*8) P    - THE NINE KNOT POINTS
C        (R*8) X    - THE POINT AT WHICH TO EVALUATE THE SPLINE
C  OUTPUT:
C        (R*8) SP9  - THE EVALUATED SPLINE
C
C  ROUTINES: NONE
C
C  DATE:     24-05-99 VERSION 1.1
C  AUTHOR:   HUGH SUMMERS, UNIVERSITY OF STRATHCLYDE
C            FIRST RELEASE
C
C
C-----------------------------------------------------------------------
      REAL*8              P(NSPLINE),  X
```

2.65 spfman11: Subroutine spfman11 from library adas1xx

```
      SUBROUTINE SPFMAN11(Z0,Z,Z1,IIGRP0,IRGRP0,TITLE,
     &     ACIA,BCIA,ACRA,BCRA,
     &     ISHE,IRES,INS1,ILS1,ES1,IZS1,INS2,
     &     ILS2,ES2,IZS2,ER1,ER2,WR1,WR2,
     &     IIFTYP,IIOTYP,IIFOUT,IXMAX,ITMAX,XDAT,EDAT,TDAT,
     &     IOP,ASCL,
     &     XA, YA, APA, XP, YP, TOA, YOA, YOAP,
     &     NITHR, NRTHR , CIA , CRA, NA , LA ,
     &     EIONA , IZETAA , ENERA , WGHTA, EMIN)
       IMPLICIT REAL*8(A-H,O-Z)
C-----------------------------------------------------------------------
C
C  ************  FORTRAN 77 PROGRAM:  SPFMAN11 ************************
C
C
C  PURPOSE: FIT IONISATION CROSS-SECTION DATA WITH SIMPLE APPROXIMATE
C  FORMS AND TO CALCULATE IONISATION RATE COEFFICIENTS
C
C  THE APPROXIMATE FORM ADOPTED IS A SUM OF TERMS ARISING FROM EACH
C  SHELL OF THE BCHID TYPE + A SUM OF TERMS OF EXCITATION CROSS-SECTION
C  FORM. THE LATTER SEEK TO REPRESENT SHARP ABOVE THRESHOLD AUTOIONISING
C  FEATURES. SCALING FACTORS ARE ASSIGNED TO AT MOST TWO SHELL GROUPS
C  (A GROUP FOR EXAMPLE BEING L-SHELLS OF THE SAME N) AND TWO RESONANCE
C  GROUPS. THE SCALING FACTORS ARE OBTAINED BY LEAST SQUARE FITTING TO
C  THE OBSERVED DATA.
C
C  DATA:
C         THIS PROGRAM IS NOT YET PROPERLY ANNOTATED
C
C  INPUT:
C     (R*8)  Z0     = NUCLEAR CHARGE OF ION
C     (R*8)  Z      = INITIAL ION CHARGE
C     (R*8)  Z1     = FINAL ION CHARGE
C     (I*4)  IIGRP0 = NO. OF SHELL GROUPS
C     (I*4)  IRGRP0 = NO. OF RESONANCE GROUPS
C     (C*40) TITLE  = TITLE FOR THIS RUN
C     (R*8)  ACIA   = SCALING PARAMETER FOR SHELL GROUP 1
C     (R*8)  BCIA   = SCALING PARAMETER FOR SHELL GROUP 2
C     (R*8)  ACRA   = SCALING PARAMETER FOR RESONANCE GROUP 1
C     (R*8)  BCRA   = SCALING PARAMETER FOR RESONANCE GROUP 2
C     (I*4)  ISHE() = NO. OF ENTRIES FOR EACH SHELL GROUP (1-6)
C     (I*4)  IRES() = NO. OF ENTRIES FOR EACH RESONANCE GROUP (1-6)
C     (I*4)  INS1() = SHELL GROUP 1 DATA: N
C     (I*4)  ILS1() = SHELL GROUP 1 DATA: L
C     (R*8)  ES1()  = SHELL GROUP 1 DATA: EION(RYD)
C     (I*4)  IZS1() = SHELL GROUP 1 DATA: IZETA
C     (I*4)  INS2() = SHELL GROUP 2 DATA: N
C     (I*4)  ILS2() = SHELL GROUP 2 DATA: L
C     (R*8)  ES2()  = SHELL GROUP 2 DATA: EION(RYD)
C     (I*4)  IZS2() = SHELL GROUP 2 DATA: IZETA
C     (R*8)  ER1()  = RESONANCE GROUP 1 DATA: ENERGY(RYD)
C     (R*8)  ER2()  = RESONANCE GROUP 2 DATA: ENERGY(RYD)
C     (R*8)  WR1()  = RESONANCE GROUP 1 DATA: WEIGHT
C     (R*8)  WR2()  = RESONANCE GROUP 2 DATA: WEIGHT
C     (I*4)  IIFTYP = ENERGY PARAMETER FORM
C                       1 : INCIDENT ENERGY (RYD)
C                       2 : INCIDENT ENERGY (EV)
C                       3 : X THRESHOLD PARAMETER
C     (I*4)  IIOTYP = CROSS-SECTIONAL FORM
```

```
C                     1 : X-SECT. (PI*A0**2)
C                     2 : X-SECT. (CM**2)
C                     3 : COLLISION STRENGTH (OMEGA)
C                     4 : SCALED COLLISION STRENGTH ((Z**2)*OMEGA)
C     (I*4)  IIFOUT = OUTPUT TEMPERATURE FORM
C                     1 : KELVIN
C                     2 : EV
C                     3 : SCALED UNITS (TE(K)/(Z1**2))
C     (I*4)  IXMAX  = NUMBER OF X-SECT./ENERGY PAIRS
C     (I*4)  ITMAX  = NUMBER OF TEMPS.
C     (R*8)  XDAT() = X-SECTION DATA
C     (R*8)  EDAT() = ENERGY DATA
C     (R*8)  TDAT() = TEMPERATURE DATA
C     (I*4)  IOP    = USE DEFAULT SCALING PARAMS? (1 = YES, 0 = NO)
C     (R*8)  ASCL  = GRAPHIC SCALING PARAMETER
C
C  OUTPUT:
C     (R*8)  XA()   = SCALED ENERGY
C     (R*8)  YA()   = OMEGA
C     (R*8)  APA()  = APPROXIMATE OMEGA
C     (R*4)  XP()   = SCALED ENERGY RESONANCE POINTS
C     (R*4)  YP()   = OMEGA OF RESONANCE POINTS
C     (R*8)  TOA()  = TEMP (KELVIN)
C     (R*8)  YOA()  = S, MAXWELL AVERAGED IONISATION RATE COEFF.(CM^3 S^-1)
C     (R*8)  YOAP() = SEM, APPROXIMATE RATE COEFF.
C     (I*4)  NITHR  = NUMBER OF RESONANCES
C     (I*4)  NRTHR  = NUMBER OF EXTRA (?) RESONANCES
C     (R*8)  CIA()  = OUTPUT SCALING PARAMS
C     (R*8)  CRA()  = OUTPUT SCALING PARAMS
C     (I*4)  NA(,)  = SHELL GROUP DATA : N
C     (I*4)  LA(,)  = SHELL GROUP DATA : L
C     (R*8)  EIONA(,) = SHELL GROUP DATA : EION(RYD)
C     (I*4)  IZETAA(,)= SHELL GROUP DATA : IZETA
C     (R*8)  ENERA(,) = RESONANCE GROUP DATA : ENERGY(RYD)
C     (R*8)  WGHTA(,) = RESONANCE GROUP DATA : WEIGHT
C
C  AUTHOR:
C
C  ************ H.P.SUMMERS, JET         1 JULY 1987  *****************
C  ************ J.SPENCE, JET              JULY 1987  *****************
C  *** COR                              30 OCT  1989               ***
C
C  UPDATE :  30/4/92 W DICKSON  -  ADD PARAMETER NDTEM AND SET TO 30
C                                  ALTER OUTPUT TEMP PANEL ACCORDINGLY
C
C  UPDATE :   7/5/92 W DICKSON  -  ADJUST OUTPUT TO INCLUDE TEMPERATURE
C                                  IN EV, AND FORMAT AS SZD FILE
C
C-----------------------------------------------------------------------
C
C UNIX-IDL CONVERSION:
C
C VERSION: 1.1                          DATE: 23-08-96
C MODIFIED: WILLIAM OSBORN
C               - FIRST CONVERTED.
C               - MADE INTO A SUBROUTINE. COMMENTED-OUT PANEL INPUT
C                 AND GRAPHICAL ROUTINES.
C
C VERSION: 1.2                          DATE: 03-10-96
C MODIFIED: WILLIAM OSBORN
C               - ALLOWED ISTOP = 5 FROM XXSIM TO GIVE NO ERROR
```

```
C
C VERSION: 1.3                          DATE: 08-10-96
C MODIFIED: WILLIAM OSBORN
C               - LET YAT BE YA WEIGHTED BY X1 TO AGREE WITH NAG VERSION,
C                 IT WAS UNWEIGHTED
C
C VERSION: 1.4                          DATE: 08-10-96
C MODIFIED: WILLIAM OSBORN
C               - ADDED NA AND LA TO PARAMETERS
C
C VERSION 1.5 DATE: 14/02/97
C MODIFIED: RICHARD MARTIN
C     - ADDED STATEMENTS '136 CONTINUE' AND '146 CONTINUE'
C         IMMEDIATELY AFTER '135 CONTINUE' AND '145 CONTINUE'
C     - CHANGED 'GOTO 135' TO 'GOTO 136' ETC..
C
C VERSION: 1.6 DATE: 15/05/07
C MODIFIED:  Allan Whiteford
C               - Updated comments as part of subroutine
C                 documentation production.
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      CHARACTER*40        TITLE
      INTEGER             IIFOUT,      IIFTYP,      IIGRP0,      IIOTYP
      INTEGER             ILS1(6),     ILS2(6),     INS1(6),     INS2(6)
      INTEGER             IOP,         IRES(2),     IRGRP0,      ISHE(2)
      INTEGER             ITMAX,       IXMAX,       IZETAA(6,2), IZS1(6)
      INTEGER             IZS2(6),     LA(6,2),     NA(6,2),     NITHR
      INTEGER             NRTHR
      REAL*8              ACIA,        ACRA,        APA(40),     ASCL
      REAL*8              BCIA,        BCRA,        CIA(2),      CRA(2)
      REAL*8              EDAT(40),    EIONA(6,2),  EMIN
      REAL*8              ENERA(6,2),  ER1(6),      ER2(6),      ES1(6)
      REAL*8              ES2(6),      TDAT(NDTEM), TOA(NDTEM)
      REAL*8              WGHTA(6,2),  WR1(6),      WR2(6),      XA(40)
      REAL*8              XDAT(40)
      REAL                XP(12)
      REAL*8              YA(40),      YOA(NDTEM),  YOAP(NDTEM)
      REAL                YP(12)
      REAL*8              Z,           Z0,          Z1
```

2.66 spfman12: Subroutine spfman12 from library adas1xx

```
      SUBROUTINE SPFMAN12(Z0,Z,Z1,IIGRP0,IRGRP0,TITLE,
     &     ACIA,BCIA,ACRA,BCRA,
     &     ISHE,IRES,INS1,ILS1,ES1,IZS1,INS2,
     &     ILS2,ES2,IZS2,ER1,ER2,WR1,WR2,
     &     IIFTYP,IIOTYP,IIFOUT,IXMAX,ITMAX,XDAT,EDAT,TDAT,
     &     IOP,ASCL,
     &     XA, YA, APA, XP, YP, TOA, YOA, YOAP,
     &     NITHR, NRTHR , CIA , CRA,
     &     NA , LA , EIONA , IZETAA , ENERA , WGHTA, EMIN)
      IMPLICIT REAL*8(A-H,O-Z)
C-----------------------------------------------------------------------
C
C  ************  FORTRAN 77 PROGRAM:  SPFMAN12 ************************
C
C
C  PURPOSE: FIT IONISATION RATE COEFFT. DATA WITH SIMPLE APPROXIMATE
C  FORMS AND TO CALCULATE RATE AT DIFFERENT TEMPERATURES.
C
C  THE APPROXIMATE FORM ADOPTED IS A SUM OF TERMS ARISING FROM EACH
C  SHELL OF THE BCHID TYPE + A SUM OF TERMS OF EXCITATION CROSS-SECTION
C  FORM. THE LATTER SEEK TO REPRESENT SHARP ABOVE THRESHOLD AUTOIONISING
C  FEATURES. SCALING FACTORS ARE ASSIGNED TO AT MOST TWO SHELL GROUPS
C  (A GROUP FOR EXAMPLE BEING L-SHELLS OF THE SAME N) AND TWO RESONANCE
C  GROUPS. THE SCALING FACTORS ARE OBTAINED BY LEAST SQUARE FITTING TO
C  THE OBSERVED DATA.
C
C  DATA:
C         THIS PROGRAM IS NOT YET PROPERLY ANNOTATED
C
C  INPUT:
C     (R*8)  Z0     = NUCLEAR CHARGE OF ION
C     (R*8)  Z      = INITIAL ION CHARGE
C     (R*8)  Z1     = FINAL ION CHARGE
C     (I*4)  IIGRP0 = NO. OF SHELL GROUPS
C     (I*4)  IRGRP0 = NO. OF RESONANCE GROUPS
C     (C*40) TITLE  = TITLE FOR THIS RUN
C     (R*8)  ACIA   = SCALING PARAMETER FOR SHELL GROUP 1
C     (R*8)  BCIA   = SCALING PARAMETER FOR SHELL GROUP 2
C     (R*8)  ACRA   = SCALING PARAMETER FOR RESONANCE GROUP 1
C     (R*8)  BCRA   = SCALING PARAMETER FOR RESONANCE GROUP 2
C     (I*4)  ISHE() = NO. OF ENTRIES FOR EACH SHELL GROUP (1-6)
C     (I*4)  IRES() = NO. OF ENTRIES FOR EACH RESONANCE GROUP (1-6)
C     (I*4)  INS1() = SHELL GROUP 1 DATA: N
C     (I*4)  ILS1() = SHELL GROUP 1 DATA: L
C     (R*8)  ES1()  = SHELL GROUP 1 DATA: EION(RYD)
C     (I*4)  IZS1() = SHELL GROUP 1 DATA: IZETA
C     (I*4)  INS2() = SHELL GROUP 2 DATA: N
C     (I*4)  ILS2() = SHELL GROUP 2 DATA: L
C     (R*8)  ES2()  = SHELL GROUP 2 DATA: EION(RYD)
C     (I*4)  IZS2() = SHELL GROUP 2 DATA: IZETA
C     (R*8)  ER1()  = RESONANCE GROUP 1 DATA: ENERGY(RYD)
C     (R*8)  ER2()  = RESONANCE GROUP 2 DATA: ENERGY(RYD)
C     (R*8)  WR1()  = RESONANCE GROUP 1 DATA: WEIGHT
C     (R*8)  WR2()  = RESONANCE GROUP 2 DATA: WEIGHT
C     (I*4)  IIFTYP = ENERGY PARAMETER FORM
C                       1 : INCIDENT ENERGY (RYD)
C                       2 : INCIDENT ENERGY (EV)
C                       3 : X THRESHOLD PARAMETER
C     (I*4)  IIOTYP = CROSS-SECTIONAL FORM
```

```
C                     1 : X-SECT. (PI*A0**2)
C                     2 : X-SECT. (CM**2)
C                     3 : COLLISION STRENGTH (OMEGA)
C                     4 : SCALED COLLISION STRENGTH ((Z**2)*OMEGA)
C     (I*4)  IIFOUT = OUTPUT TEMPERATURE FORM
C                     1 : KELVIN
C                     2 : EV
C                     3 : SCALED UNITS (TE(K)/(Z1**2))
C     (I*4)  IXMAX  = NUMBER OF X-SECT./ENERGY PAIRS
C     (I*4)  ITMAX  = NUMBER OF TEMPS.
C     (R*8)  XDAT() = X-SECTION DATA
C     (R*8)  EDAT() = ENERGY DATA
C     (R*8)  TDAT() = TEMPERATURE DATA
C     (I*4)  IOP    = USE DEFAULT SCALING PARAMS? (1 = YES, 0 = NO)
C     (R*8)  ASCL  = GRAPHIC SCALING PARAMETER
C
C  OUTPUT:
C     (R*8)  XA()   = SCALED ENERGY
C     (R*8)  YA()   = OMEGA
C     (R*8)  APA()  = APPROXIMATE OMEGA
C     (R*4)  XP()   = SCALED ENERGY RESONANCE POINTS
C     (R*4)  YP()   = OMEGA OF RESONANCE POINTS
C     (R*8)  TOA()  = TEMP (KELVIN)
C     (R*8)  YOA()  = S, MAXWELL AVERAGED IONISATION RATE COEFF.(CM^3 S^-1)
C     (R*8)  YOAP() = SEM, APPROXIMATE RATE COEFF.
C     (I*4)  NITHR  = NUMBER OF RESONANCES
C     (I*4)  NRTHR  = NUMBER OF EXTRA (?) RESONANCES
C     (R*8)  CIA()  = OUTPUT SCALING PARAMS
C     (R*8)  CRA()  = OUTPUT SCALING PARAMS
C     (I*4)  NA(,)  = SHELL GROUP DATA : N
C     (I*4)  LA(,)  = SHELL GROUP DATA : L
C     (R*8)  EIONA(,) = SHELL GROUP DATA : EION(RYD)
C     (I*4)  IZETAA(,)= SHELL GROUP DATA : IZETA
C     (R*8)  ENERA(,) = RESONANCE GROUP DATA : ENERGY(RYD)
C     (R*8)  WGHTA(,) = RESONANCE GROUP DATA : WEIGHT
C
C  AUTHOR:
C  ************ H.P.SUMMERS, JET         5 FEB 1989   *****************
C  *** COR                              30 OCT 1989                ***
C
C-----------------------------------------------------------------------
C
C UNIX-IDL CONVERSION:
C
C VERSION: 1.1                          DATE: 07-10-96
C MODIFIED: WILLIAM OSBORN
C               - FIRST CONVERTED.
C               - MADE INTO A SUBROUTINE. COMMENTED-OUT PANEL INPUT
C                 AND GRAPHICAL ROUTINES.
C
C VERSION: 1.2                          DATE: 15-05-07
C MODIFIED: Allan Whiteford
C               - Updated comments as part of subroutine documentation
C                 project.
C-----------------------------------------------------------------------
      CHARACTER*40       TITLE
      INTEGER            IIFOUT,      IIFTYP,      IIGRP0,      IIOTYP
      INTEGER            ILS1(6),     ILS2(6),     INS1(6),     INS2(6)
      INTEGER            IOP,         IRES(2),     IRGRP0,      ISHE(2)
      INTEGER            ITMAX,       IXMAX,       IZETAA(6,2), IZS1(6)
      INTEGER            IZS2(6),     LA(6,2),     NA(6,2),     NITHR
```

```
INTEGER            NRTHR
REAL*8             ACIA,         ACRA,         APA(40),      ASCL
REAL*8             BCIA,         BCRA,         CIA(2),       CRA(2)
REAL*8             EDAT(40),     EIONA(6,2),   EMIN
REAL*8             ENERA(6,2),   ER1(6),       ER2(6),       ES1(6)
REAL*8             ES2(6),       TDAT(NDTEM),  TOA(NDTEM)
REAL*8             WGHTA(6,2),   WR1(6),       WR2(6),       XA(40)
REAL*8             XDAT(40)
REAL               XP(12)
REAL*8             YA(40),       YOA(NDTEM),   YOAP(NDTEM)
REAL               YP(12)
REAL*8             Z,            Z0,           Z1
```

2.67 spfman5e: Subroutine spfman5e from library adas1xx

```
      SUBROUTINE SPFMAN5E( Z0    , Z     , ZEFF  , TITLE , IETYP ,
     &                     IXTYP , IND1  , IND2  , WI, WJ, EI, EJ,
     &                     IATYP , ACOEFF, IFTYP , IOTYP , IFOUT ,
     &                     IXMAX , ITMAX , EDAT  , XDAT  , TDAT  ,
     &                     IIORD , IIBTS , IIFPT , IIXOP , IIDIF ,
     &                     XTIT1 , IGRD1 , XL1   , XU1   ,
     &                     YL1   , YU1   , XTIT2 , IGRD2 ,
     &                     XL2   , XU2   , YL2   , YU2   , IWRITE,
     &                     FXC2  , FXC3  , XA    , YA    , APOMA ,
     &                     DIFOMA, TOA   , GOA   , APGOA , EXCRA ,
     &                     DEXCRA, GBARFA,
     &                     ICT   , ITOUT , S     , FIJ   , EIJ
     &                    )
      IMPLICIT REAL*8(A-H,O-Z)
C-----------------------------------------------------------------------
C  PURPOSE: ANALYSE ELECTRON IMPACT RATE DATA AND CONVERT TO
C  RATE COEFFICIENTS
C
C  VARIOUS FORMS OF DATA ENTRY ARE ALLOWED
C
C  EXTENDED ARRAY DIMENSION VERSION OF SPFMAIN5
C
C  DATA IS FITTED WITH APPROXIMATE FORMS TO AID INTERPOLATION DEPENDING
C  ON THE TRANSITION TYPE. THESE ARE
C                1. DIPOLE
C                2. NON-DIPOLE
C                3. SPIN CHANGE
C                4. OTHER
C
C  DATA ENTRY IS VIA CALL TO PANEL SUBROUTINE  SPFMA4E  AS FOLLOWS:
C
C  INPUT
C      IPAN = INITIAL PANEL NUMBER AT START
C
C  OUTPUT
C      IPAN   = FINAL PANEL NUMBER
C      ANS    = YES  - FINISH UP CALCULATION SINCE NO MORE CASES
C             = NO   - DATA FOR NEW CASE RETURNED
C      Z0     = NUCLEAR CHARGE OF ION
C      Z      = ION CHARGE
C      ZEFF   = ION CHARGE + 1
C      TITLE  = TITLE FOR CASE
C      IETYP  = 1  LEVEL ENERGIES IN CM-1
C             = 2  LEVEL ENERGIES IN RYD
C      IXTYP  = 1  DIPOLE TRANSITION
C             = 2  NON-DIPOLE TRANSITION
C             = 3  SPIN CHANGE TRANSITION
C             = 4  OTHER
C      IND1   = LOWER LEVEL INDEX  (USER CHOICE)
C      IND2   = UPPER LEVEL INDEX  (USER CHOICE)
C      WI     = LOWER LEVEL STATISTICAL WEIGHT
C      WJ     = UPPER LEVEL STATISTICAL WEIGHT
C      EI     = LOWER LEVEL ENERGY (IN SELECTED UNITS)
C      EJ     = UPPER LEVEL ENERGY
C      IATYP  = 1  A-COEFFICIENT RETURNED
C             = 2  OSCILLATOR STRENGTH RETURNED
C             = 3  LINE STRENGTH RETURNED
C      ACOEFF = TRANSITION PROBABILITY (IN ABOVE FORM, DIPOLE CASE ONLY)
C      IFTYP  = 1 KELVIN                    FOR SOURCE TEMP. UNITS
```

```
C             = 2  EV                          FOR SOURCE TEMP. UNITS
C             = 3 SCALED UNITS (TE(K)/Z1**2) FOR SOURCE TEMP. UNITS
C             = 4 REDUCED UNITS (KTE/EIJ)    FOR SOURCE TEMP. UNITS
C      IOTYP  = 1 EXCITATION RATE COEFFICIENT (CM3 S-1) RETURNED
C             = 2 DE-EXCITATION RATE COEFFICIENT (CM3 S-1) RETURNED
C             = 3 UPSILON RETURNED
C      IFOUT  = 1 KELVIN FOR OUTPUT TEMPERATURE UNIT
C             = 2   EV   FOR OUTPUT TEMPERATURE UNIT
C             = 3 SCALED UNITS  (TE(K)/Z1**2)
C             = 4 REDUCED UNITS RETURNED (KTE/EIJ)
C      IXMAX  = NUMBER OF TEMP/RATE PAIRS ENTERED
C      ITMAX  = NUMBER OF OUTPUT TEMPERATURES ENTERED
C      EDAT(I)= INPUT TEMPS.         (SELECTED UNITS)
C      XDAT(I)= INPUT RATE COEFFTS.  (SELECTED UNITS)
C      TDAT(I)= OUTPUT TEMPS.        (SELECTED UNITS)
C      IIORD  = *** UNUSED ***
C      IIGPH  = 0  NO COMPARARIVE GRAPH TO BE PRODUCED
C             = 1     COMPARATIVE GRAPH TO BE PRODUCED
C      IIGPG  = 0  NO GAMMA  GRAPH TO BE PRODUCED
C             = 1     GAMMA  GRAPH TO BE PRODUCED
C      IIBTS  = 0  BAD POINT OPTION OFF
C             = 1  BAD POINT OPTION ON
C      IIFPT  = 1  SELECT ONE POINT OPTIMISING
C             = 2  SELECT TWO POINT OPTIMISING
C      IIXOP  = 0  OPTIMISING OFF
C             = 1  OPTIMISING ON  (IF ALLOWED)
C      IIDIF  = *** UNUSED ***
C      XTIT1  = SPECIFIC TITLE FOR COMPARATIVE GRAPH
C      IGRD1  = 10 DO NOT PUT GRAPH IN A GRIDFILE
C             = 11 PUT GRAPH IN A GRIDFILE
C      IDEF1  = 11 USE DEFAULT SCALING FOR GRAPH
C             = 10 SCALING FOR GRAPH RETURNED
C      XL1    = LOWER X FOR COMPARATIVE GRAPH
C      XU1    = UPPER X FOR COMPARARIVE GRAPH
C      YL1    = LOWER Y FOR COMPARATIVE GRAPH
C      YU1    = UPPER Y FOR COMPARATIVE GRAPH
C      XTIT2  = SPECIFIC TITLE FOR GAMMA GRAPH
C      IGRD2  = 10 DO NOT PUT GRAPH IN A GRIDFILE
C             = 11 PUT GRAPH IN A GRIDFILE
C      IDEF2  = 11 USE DEFAULT SCALING FOR GRAPH
C             = 10 SCALING FOR GRAPH RETURNED
C      XL2    = LOWER X FOR GAMMA GRAPH
C      XU2    = UPPER X FOR GAMMA GRAPH
C      YL2    = LOWER Y FOR GAMMA GRAPH
C      YU2    = UPPER Y FOR GAMMA GRAPH
C
C
C  AUTHOR:  HUGH P. SUMMERS, UNIVERSITY OF STRATHCLYDE
C           TEL. 0141-553-4196
C
C  DATE:  15/11/96
C
C
C  DATE:     15-11-95 VERSION 1.1
C  MODIFIED: HUGH P. SUMMERS
C               - FIRST EDITION
C
C  DATE:     25-11-95 VERSION 1.2
C  MODIFIED: WILLIAM OSBORN
C               - CHANGED PARAMS TO PARAMS102
C  DATE:     20-05-99 VERSION 1.3
```

```
C  MODIFIED: HUGH SUMMERS
C
C  DATE:     15-05-2007 VERSION 1.4
C  MODIFIED: Allan Whiteford
C               - Updated comments as part of subroutine
C                 documentation production.
C-----------------------------------------------------------------------
       INCLUDE 'PARAMS'
C-----------------------------------------------------------------------
      CHARACTER*40        TITLE,       XTIT1,       XTIT2
      INTEGER             IATYP,       ICT,         IETYP,       IFOUT
      INTEGER             IFTYP,       IGRD1,       IGRD2,       IIBTS
      INTEGER             IIDIF,       IIFPT,       IIORD,       IIXOP
      INTEGER             IND1,        IND2,        IOTYP,       ITMAX
      INTEGER             ITOUT,       IWRITE,      IXMAX,       IXTYP
      REAL*8              ACOEFF,      APGOA(ISTDIM)
      REAL*8              APOMA(ISTDIM),            DEXCRA(ISTDIM)
      REAL*8              DIFOMA(ISTDIM),           EDAT(ISTDIM)
      REAL*8              EI,          EIJ,         EJ
      REAL*8              EXCRA(ISTDIM),            FIJ,         FXC2
      REAL*8              FXC3,        GBARFA(ISTDIM)
      REAL*8              GOA(ISTDIM), S,           TDAT(ISTDIM)
      REAL*8              TOA(ISTDIM), WI,          WJ
      REAL*8              XA(ISTDIM),  XDAT(ISTDIM),            XL1
      REAL*8              XL2,         XU1,         XU2
      REAL*8              YA(ISTDIM),  YL1,         YL2,         YU1
      REAL*8              YU2,         Z,           Z0,          ZEFF
```

2.68 spfman8hx: Subroutine spfman8hx from library adas1xx

```
      SUBROUTINE SPFMAN8HX(Z0,Z1,Z,N0,V0,PHFRAC,TITLE,
     &     IIOTYP,IIFTYP,IIFOUT,IXMAX,ITMAX,EDAT,XDAT,TDAT,
     &     INGRUP,IIATYP,IGRP,EDGRP,SCGRP,
     &     NIGRP1,LIGRP1,WIGRP1,NJGRP1,LJGRP1,WJGRP1,EGRP1,AGRP1,CRGRP1,
     &     NIGRP2,LIGRP2,WIGRP2,NJGRP2,LJGRP2,WJGRP2,EGRP2,AGRP2,CRGRP2,
     &     IGF,IBU,IFSEL,IBT,IGH,ASCL,NCGRP1,NCGRP2,
     &     XPA,YPA,ICT,XP,YP,ICOUT,
     &     TEA4,EXCRA,
     &     NGROUP, IGROUP, NIA   , LIA    , NJA   , LJA    , NCUTA ,
     &     EMEAN , EDISPO, SCALEO, EDISP  , SCALE , WIA   ,
     &     WJA   , EIJA  , FIJA  , CORFIA , CORFA ,
     &     XA    , YA    , XSAOLD, YSAOLD , YSAOLP,
     &     XSANEW, APGA  , APGAP , YSANEW , YSANWP, APGOAP,
     &     SCEXP , XOA   , YOA   , APGOA  , ITYPEA, EMEAN4, SCEXP4,
     &     EIJA4 , FIJA4 , CORFA4, EDISP4 , SCALE4)

      IMPLICIT REAL*8(A-H,O-Z)
C-----------------------------------------------------------------------
C
C  ************  FORTRAN 77 PROGRAM:  SPFMAN8H ************************
C
C  PURPOSE: GRAPH AND INTERPOLATE DIELECTRONIC RECOMBINATION
C    COEFFICIENTS.
C
C  INPUT DATA IS REQUIRED ON THE ION SPECIFICATION AND A SPECIFED SET
C    SET OF RECOMBINATION COEFFICIENTS V TEMPERATURE. SUPPLEMENTARY
C    DATA ON CORE ENERGIES AND OSCILLATOR STRENGTHS IS ALSO
C    REQUIRED.  THIS IS DIVIDED INTO TWO GROUPS, IE. DELTAN=0 AND 1.
C  AN ATTEMPT IS MADE TO REDUCE THE INPUT RECOMBINATION DATA
C    TO STANDARD ZERO DENSITY RECOMBINATION COEFFICIENTS.
C  THE COEFFICIENTS ARE THEN RATIOED TO THE BURGESS GENERAL FORMULA
C    VALUES BASED ON THE SUPPLEMENTARY INPUT DATA. THE SUBSEQUENT
C    STRATEGY DEPENDS ON WHETHER THERE THERE ARE ONE OR TWO GROUPS
C    PRESENT.
C        CASE 1. A REGRESSION LINE IS FITTED IN THE LOG(ALF/ALFGF)
C                EMEAN/TE PLANE TO OBTAIN AN ENERGY AND SCALE
C                FACTOR CORRECTIONS TO THE GENERAL FORMULA. WITH ONLY
C                ONE DATA VALUE A SCALE FACTOR IS INFERRED AND THE
C                THE ENERGY CORRECTION ASSUMED ZERO.
C        CASE 2. A SPLINE IS FITTED IN THE LOG(ALF/ALFGF) V EMEAN/TE
C                PLANE WITH ZERO CURVATURE END CONDITIONS.  AN ATTEMPT
C                IS MADE TO INTERPRET THE END SLOPES AND INTERCEPTS IN
C                TERMS OF SCALE FACTORS AND ENERGY CORRECTIONS FOR EACH
C                GROUP AS A WHOLE.
C  THE ALF/ALFGF V EMEAN/TE PLOTS ARE REFORMED OPTIONALLY USING THE
C    BURGESS GENERAL FORMULA WITH OR WITHOUT THE SCALE AND ENERGY
C    CORRECTIONS. THIS IS INTERPOLATED TO REQUIRED OUTPUT TEMPERATURES.
C  THE BURGESS GENERAL PROGRAM IS FITTED TO THE ADJUSTED BURGESS GENERAL
C    FORMULA BY A SINGLE FACTOR WHICH RAMPS THE LOW PARTIAL WAVE
C    CORRECTIONS UP OR DOWN.  NB. AN ADJUSTMENT IS ALSO MADE FOR THE
C    SPECIFIC ION FOR THE PHASE SPACE ACCESSIBLE FOR CAPTURE TO THE
C    THE LOWEST LEVEL FOR THE DELTAN=1 GROUP.  THIS IS THEN USED
C    IN BUNDLED N POPULATION CODES TO GIVE PROPER DENSITY CORRECTIONS.
C
C  DATA:
C         THIS PROGRAM IS NOT YET PROPERLY ANNOTATED
C
C  INPUT:
C     (R*8)  Z0    NUCLEAR CHARGE
```

```
C     (R*8)  Z1    RECOMBINING ION CHARGE
C     (R*8)  Z     RECOMBINED ION CHARGE
C     (I*4)  N0     = LOWEST ACCESSIBLE PRINCIPLE QUANTUM NO.
C     (R*8)  V0     = LOWEST ACCESSIBLE EFF. PRINCIPLE QUANTUM NO.
C     (R*8)  PHFRAC = LOWEST ACCESSIBLE PHASE OCCUPATION FACTOR
C     (C*40) TITLE  = TITLE FOR RUN
C     (I*4)  IIOTYP = INPUT RATE FORM, 1=DIELECTRONIC
C                                      2=DIEL. + RADIATIVE
C     (I*4)  IIFTYP = INPUT TEMP. FORM, 1= KELVIN
C                                       2= EV
C                                       3= SCALED (K/Z1**2)
C     (I*4)  IIFOUT = INPUT TEMP. FORM, 1= KELVIN
C                                       2= EV
C                                       3= SCALED (K/Z1**2)
C     (I*4)  IXMAX  = NUMBER OF TEMP./RATE PAIRS
C     (I*4)  ITMAX  = NUMBER OF OUTPUT TEMPS
C     (R*8)  EDAT() = USER OR ARCHIVE ENTERED INPUT TEMPERATURE
C     (R*8)  XDAT() = USER OR ARCHIVE ENTERED RATE
C     (R*8)  TDAT() = USER OR ARCHIVE ENTERED OUTPUT TEMPERATURE
C     (I*4)  INGRUP = NO OF CORE TRANSITION GROUPS
C     (I*4)  IIATYP = TRANSITION PROB. FORM, 1= A-COEFFICIENT
C                                            2= OSCILLATOR STRENGTH
C                                            3= LINE STRENGTH
C     (I*4)  IGRP() = NO OF ENTRIES FOR EACH GROUP
C     (R*4)  EDGRP()= INITIAL MEAN ENERGY DISP. FOR EACH GROUP
C     (R*4)  SCGRP()= MEAN SCALE FACTOR FOR EACH GROUP
C     (I*4)  NIGRP1()= TRANSITION GROUP 1 INFO
C     (I*4)  LIGRP1()= "
C     (R*8)  WIGRP1()= "
C     (I*4)  NJGRP1()= "
C     (I*4)  LJGRP1()= "
C     (R*8)  WJGRP1()= "
C     (R*8)  EGRP1()=  "
C     (R*8)  AGRP1()=  "
C     (R*8)  CRGRP1()= "
C     (I*4)  NCGRP1()= "
C     (I*4)  NIGRP2()= TRANSITION GROUP 2 INFO
C     (I*4)  LIGRP2()= "
C     (R*8)  WIGRP2()= "
C     (I*4)  NJGRP2()= "
C     (I*4)  LJGRP2()= "
C     (R*8)  WJGRP2()= "
C     (R*8)  EGRP2()=  "
C     (R*8)  AGRP2()=  "
C     (R*8)  CRGRP2()= "
C     (I*4)  NCGRP2()= "
C     (I*4)  IGF    = OPTIMISE BURGESS FORMULA FIT? 1=YES 0=NO
C     (I*4)  IBU    = OPTIMISE BURGESS PROGRAM FIT? 1=YES 0=NO
C     (I*4)  IFSEL  = 0=FIT TO FORMULA, 1=FIT TO INPUT DATA
C     (I*4)  IBT    = BAD POINT OPTION 1=ON 0=OFF
C     (I*4)  IGH    = GRAPHICAL DISPLAY? 1=YES 0=NO
C     (R*8)  ASCL   = GRAPH SCALING PARAMETER
C
C  OUTPUT:
C     (R*4)  XPA(,) = X VALUES FOR RATIO PLOT
C     (R*4)  YPA(,) = Y VALUES FOR RATIO PLOT (RATIOS OF DIEL. RECOM. COEFFS.)
C     (I*4)  ICT    = NUMBER OF POINTS ON RATIO PLOT
C     (R*4)  XP()   = X VALUES FOR RATIO PLOT SPLINE FIT
C     (R*4)  YP()   = X VALUES FOR RATIO PLOT SPLINE FIT
C     (I*4)  ICOUT  = NUMBER OF POINTS ON COEFFICIENT PLOT
C     (R*4)  TEA4() = X VALUES FOR COEFFICIENT PLOT (TEMP.)
```

```
C     (R*4)  EXCRA(,) = Y VALUES FOR COEFFICIENT PLOT (COEFFS.)
C     (R*4)  EMEAN4 = MEAN EIJ USED IN GRAPH ANNOTATION
C     (R*4)  SCEXP4 = SCALE FACTOR USED IN GRAPH ANNOTATION
C     (R*4)  EIJA4  = VALUE OF EIJ USED IN GRAPH ANNOTATION
C     (R*4)  FIJA4  = VALUE OF FIJ USED IN GRAPH ANNOTATION
C     (R*4)  CORFA4 = VALUE OF CORFA USED IN GRAPH ANNOTATION
C     (R*4)  EDISP4 = VALUE OF DISPLACEMENT USED IN GRAPH ANNOTATION
C     (R*4)  SCALE4 = VALUE OF SCALE USED IN GRAPH ANNOTATION
C     (I*4)  NGROUP = NUMBER OF CORE TRANSITION GROUPS
C     (I*4)  IGROUP()=NUMBER OF ENTRIES FOR EACH GROUP
C     (I*4)  NIA(,) = NI VALUES FOR BOTH GROUPS
C     (I*4)  LIA(,) = LI VALUES FOR BOTH GROUPS
C     (I*4)  NJA(,) = NJ VALUES FOR BOTH GROUPS
C     (I*4)  LJA(,) = LJ VALUES FOR BOTH GROUPS
C     (I*4)  NCUTA(,)= NCUT VALUES FOR BOTH GROUPS
C     (R*8)  EMEAN()= MEAN ENERGY VALUES FOR EACH GROUP
C     (R*8)  EDISPO()=INITIAL ENERGY DISPLACEMENT FOR EACH GROUP
C     (R*8)  SCALEO()=INITIAL SCALE FACTOR FOR EACH GROUP
C     (R*8)  EDISP()= FINAL ENERGY DISPLACEMENT FOR EACH GROUP
C     (R*8)  SCALE()= FINAL SCALE FACTOR FOR EACH GROUP
C     (R*8)  WIA(,) = WI VALUES FOR BOTH GROUPS
C     (R*8)  EIJA(,)= EIJ VALUES FOR BOTH GROUPS
C     (R*8)  FIJA(,)= FIJ VALUES FOR BOTH GROUPS
C     (R*8)  CORFIA(,)=INITIAL CORFAC VALUES FOR BOTH GROUPS
C     (R*8)  CORFA(,)= FINAL CORFAC VALUES FOR BOTH GROUPS
C     (R*8)  XA()   = INPUT TEMPERATURES
C     (R*8)  YA()   = INPUT ALF
C     (R*8)  XSAOLD()=INITIAL X VALUES
C     (R*8)  YSAOLD()=INITIAL ALF/ALFG VALUES
C     (R*8)  YSAOLP()=INITIAL ALF/ALFP VALUES
C     (R*8)  APGA() = INITIAL ALFG
C     (R*8)  APGAP()= INITIAL ALFP
C     (R*8)  YSANEW()=FINAL ALF/ALFG VALUES
C     (R*8)  YSANEWP()=FINAL ALF/ALFP VALUES
C     (R*8)  APGOA()= FINAL ALFG
C     (R*8)  APGOAP()=FINAL ALFP
C     (R*8)  SCEXP  = PARAMETER DETERMINING X-SCALE EXPANSION
C     (R*8)  XOA()  = USER OR ARCHIVE ENTERED OUTPUT TEMPS
C     (R*8)  YOA()  = INTERPOLATED ALF VALUES CALCULATED AT XOA TEMPS.
C     (R*8)  ITYPEA(,)= TYPE OF TRANSITION (USED TO CALCULATE CORRECTION
C                                          FACTORS)
C
C  AUTHOR:
C
C  *******   H.P.SUMMERS, JET           10 JUNE 1987  ***********
C  S.P.F PANEL STRUCTURE VERSION.
C  *******   J. SPENCE, JET             30 JUNE 1987  ***********
C  S.P.F. PANEL ALTERATION TO INPUT DEFAULT CORFAC
C  *******   H.P.SUMMERS, JET           17 MAY 1988   ***********
C
C-----------------------------------------------------------------------
C
C UNIX-IDL CONVERSION:
C
C VERSION: 1.1                          DATE: 01-11-96
C MODIFIED: WILLIAM OSBORN
C               - FIRST CONVERTED.
C               - MADE INTO A SUBROUTINE. COMMENTED-OUT PANEL INPUT
C                 AND GRAPHICAL ROUTINES.
C
C  VERSION 1.2                                          DATE:  08/11/96
```

```
C  MODIFIED: WILLIAM OSBORN
C               - REMOVED UNUSED VARIABLES
C
C  VERSION: 1.3                                          DATE:  15/05/07
C  MODIFIED:  Allan Whiteford
C               - Updated comments as part of subroutine
C                 documentation production.
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      CHARACTER*40        TITLE
      INTEGER             IBT,          IBU,          ICOUT,       ICT
      INTEGER             IFSEL,        IGF,          IGH
      INTEGER             IGROUP(2),    IGRP(2),      IIATYP,      IIFOUT
      INTEGER             IIFTYP,       IIOTYP,       INGRUP,      ITMAX
      INTEGER             ITYPEA(2,6), IXMAX,         LIA(2,6)
      INTEGER             LIGRP1(6),    LIGRP2(6),    LJA(2,6)
      INTEGER             LJGRP1(6),    LJGRP2(6),    N0
      INTEGER             NCGRP1(6),    NCGRP2(6),    NCUTA(2,6),  NGROUP
      INTEGER             NIA(2,6),     NIGRP1(6),    NIGRP2(6)
      INTEGER             NJA(2,6),     NJGRP1(6),    NJGRP2(6)
      REAL*8              AGRP1(6),     AGRP2(6),     APGA(10)
      REAL*8              APGAP(10),    APGOA(10),    APGOAP(10),  ASCL
      REAL*8              CORFA(2,6)
      REAL                CORFA4(2,6)
      REAL*8              CORFIA(2,6), CRGRP1(6),    CRGRP2(6)
      REAL*8              EDAT(10),     EDGRP(2),     EDISP(2)
      REAL                EDISP4(2)
      REAL*8              EDISPO(2),    EGRP1(6),     EGRP2(6)
      REAL*8              EIJA(2,6)
      REAL                EIJA4(2,6)
      REAL*8              EMEAN(2)
      REAL                EMEAN4,       EXCRA(10,15)
      REAL*8              FIJA(2,6)
      REAL                FIJA4(2,6)
      REAL*8              PHFRAC,       SCALE(2)
      REAL                SCALE4(2)
      REAL*8              SCALEO(2),    SCEXP
      REAL                SCEXP4
      REAL*8              SCGRP(2),     TDAT(10)
      REAL                TEA4(10)
      REAL*8              V0,           WIA(2,6),     WIGRP1(6)
      REAL*8              WIGRP2(6),    WJA(2,6),     WJGRP1(6)
      REAL*8              WJGRP2(6),    XA(10),       XDAT(10),    XOA(10)
      REAL                XP(10),       XPA(4,10)
      REAL*8              XSANEW(10),  XSAOLD(10),   YA(10),      YOA(10)
      REAL                YP(10),       YPA(4,10)
      REAL*8              YSANEW(10),  YSANWP(10),   YSAOLD(10)
      REAL*8              YSAOLP(10),  Z,            Z0,          Z1
```

2.69 spls: Subroutine spls from library adas1xx

```
      SUBROUTINE SPLS(XIN,Y)
C-----------------------------------------------------------------------
C  ****************** FORTRAN 77 SUBROUTINE SPLS *********************
C
C  PURPOSE:
C         COMPUTE 5-POINTS CUBIC SPLINE COEFFICIENTS
C  INPUT:
C         (R*8) XIN - THE POINT WHERE THE SPLINE IS TO BE EVALUATED
C
C  OUTPUT:
C         (R*8) Y - THE BURGESS KNOT POINTS
C
C  ROUTINES:
C         NONE
C
C  DATE:    02-07-95 VERSION 1.1
C
C  WRITTEN: A.LANZAFAME & D.H. BROOKS, UNIV.OF STRAYTHCLYDE,
C           CONVERTED FROM BURGESS BBC BASIC
C
C  DATE:    13-05-99 VERSION 1.2
C  AUTHOR:  HUGH SUMMERS, UNIV.OF STRAYTHCLYDE,
C           CORRECTED MODIFICATION OF INPUT PARAMETER X ON RETURN
C
C-----------------------------------------------------------------------
      REAL*8              XIN,         Y(5)
```

2.70 spls9: Subroutine spls9 from library adas1xx

```
      SUBROUTINE SPLS9(XIN,Y)
C-----------------------------------------------------------------------
C  ****************** FORTRAN 77 SUBROUTINE SPLS9 *******************
C
C  PURPOSE:
C         COMPUTE 9-POINTS CUBIC SPLINE COEFFICIENTS
C  INPUT:
C         (R*8) XIN - THE POINT WHERE THE SPLINE IS TO BE EVALUATED
C
C  OUTPUT:
C         (R*8) Y - THE BURGESS KNOT MULTIPIERS
C
C  ROUTINES:
C         NONE
C
C  DATE:    26-05-99 VERSION 1.1
C  AUTHOR:  HUGH SUMMERS, UNIV.OF STRAYTHCLYDE,
C
C-----------------------------------------------------------------------
      REAL*8              XIN,         Y(9)
```

2.71 three: Subroutine three from library adas1xx

```
       SUBROUTINE THREE( A , B )
C-----------------------------------------------------------------------
C
C  ******************** FORTRAN 77SUBROUTINE: THREE*********************
C
C  PURPOSE:
C        TO CALCULATE THE BURGESS KNOT POINTS
C
C  CALLING PROGRAM:
C        LSTSQ
C
C  INPUT:
C        (R*8) A - MATRIX CONTAINING REDUCED PARAMETERS
C  OUTPUT:
C        (R*8) B - THE BURGESS KNOT POINTS
C
C  ROUTINES:
C        MATIN1
C
C  DATE:   21/06/95 VERSION 1.1
C
C  AUTHOR: A.LANZAFAME (UNIV. OF STRATHCLYDE) & D.H.BROOKS
C          CONVERTED FROM BURGESS BBC BASIC
C
C-----------------------------------------------------------------------
      REAL*8             A(5,5),      B(5)
```

2.72 three9: Subroutine three9 from library adas1xx

```
      SUBROUTINE THREE9(A,P)
C-----------------------------------------------------------------------
C
C  ******************* FORTRAN 77 SUBROUTINE: THREE9  ******************
C
C  PURPOSE:
C        TO CALCULATE THE BURGESS KNOT POINTS
C
C  CALLING PROGRAM:
C        LSTSQ9
C
C  INPUT:
C        (R*8) A - MATRIX CONTAINING REDUCED PARAMETERS
C  OUTPUT:
C        (R*8) P - THE BURGESS KNOT POINTS
C
C  ROUTINES:
C        MATIN1
C
C  DATE:   26/05/99 VERSION 1.1
C
C  AUTHOR: HUGH SUMMERS, UNIVERSITY OF STRATHCLYDE
C
C-----------------------------------------------------------------------
      REAL*8              A(9,9),      P(9)
```

2.73 two: Subroutine two from library adas1xx

```
      SUBROUTINE TWO(A,B)
C-----------------------------------------------------------------------
C
C  ******************** FORTRAN 77SUBROUTINE: TWO  *********************
C
C  PURPOSE:
C        TO CALCULATE THE BURGESS KNOT POINTS
C
C  CALLING PROGRAM:
C        LSTSQ
C
C  INPUT:
C        (R*8) A - MATRIX CONTAINING REDUCED PARAMETERS
C  OUTPUT:
C        (R*8) B - THE BURGESS KNOT POINTS
C
C  ROUTINES:
C        MATIN1
C
C  DATE:   21/06/95 VERSION 1.1
C
C  AUTHOR: A.LANZAFAME (UNIV. OF STRATHCLYDE) & D.H.BROOKS
C          CONVERTED FROM BURGESS BBC BASIC
C
C-----------------------------------------------------------------------
      REAL*8             A(5,5),      B(5)
```

2.74 two9: Subroutine two9 from library adas1xx

```
      SUBROUTINE TWO9(A,P)
C-----------------------------------------------------------------------
C
C  ******************* FORTRAN 77 SUBROUTINE: TWO9  *********************
C
C  PURPOSE:
C        TO CALCULATE THE BURGESS KNOT POINTS
C
C  CALLING PROGRAM:
C        LSTSQ9
C
C  INPUT:
C        (R*8) A - MATRIX CONTAINING REDUCED PARAMETERS
C  OUTPUT:
C        (R*8) P - THE BURGESS KNOT POINTS
C
C  ROUTINES:
C        MATIN1
C
C  DATE:   26/05/99 VERSION 1.1
C
C  AUTHOR: HUGH SUMMERS, UNIVERSITY OF STRATHCLYDE
C
C-----------------------------------------------------------------------
      REAL*8              A(9,9),      P(9)
```

2.75 ups: Subroutine ups from library adas1xx

```
       FUNCTION UPS(T,X,W,IT,E,C,P,C1,C2)
C-----------------------------------------------------------------------
C
C  ******************** FORTRAN 77 FUNCTION: UPS   *********************
C
C  PURPOSE:
C        TO CALCULATE UPSILONS.
C
C  CALLING PROGRAM:
C        MAXWELL
C
C  INPUT:
C        (R*8)    T - TEMPERATURES
C        (R*8)    X - FIXED QUADRATURE POINTS
C        (R*8)    W - FIXED QUADRATURE POINTS
C        (I)     IT - TRANSITION TYPE
C        (R*8)    E - EXCITATION ENERGY
C        (R*8)    C - OMEGA SCALING PARAMETER
C        (R*8)    P - BURGESS KNOTS
C        (R*8)   C1 - SCALING PARAMETERS
C        (R*8)   C2 - SCALING PARAMTERES
C
C  OUTPUT:
C        (R*8)  UPS - UPSILONS
C
C
C  ROUTINES: NONE
C
C  DATE:     21/06/95 VERSION 1.1
C  AUTHOR:   A.LANZAFAME (UNIV. OF STRATHCLYDE) & D.H.BROOKS
C            CONVERTED FROM BURGESS BBC BASIC
C
C-----------------------------------------------------------------------
      INTEGER             IT
      REAL*8              C,           C1,          C2,          E
      REAL*8              P(5),        T,           W(K),        X(K)
```

2.76 ups9: Subroutine ups9 from library adas1xx

```
       FUNCTION UPS9( T   , IT  , E   , E0  , B   , C   ,
     &                P   , C1  , C2  ,
     &                KT  , XT  , WT  , K   , X   , W
     &              )
C-----------------------------------------------------------------------
C
C  ******************** FORTRAN 77 FUNCTION: UPS9  *********************
C
C  PURPOSE:
C        TO CALCULATE UPSILONS.
C
C  CALLING PROGRAM:
C        MAXW9
C
C  INPUT:
C        (R*8)    T    = TEMPERATURE (RYD. ENERGY UNITS)
C        (I*4)    IT   = TRANSITION TYPE
C        (R*8)    E    = EXCITATION ENERGY
C        (R*8)    E0   = SWITCHING ENERGY BETWEEN THRES. AND ASYMP.
C        (R*8)    B    = BURGESS/SUMMERS SCALING PARAMETE - B
C        (R*8)    C    = BURGESS/SUMMERS SCALING PARAMETE - C
C        (R*8)    P(9) = VALUES AT SPLINE KNOTS
C        (R*8)    C1   = SCALING PARAMETERS
C        (R*8)    C2   = SCALING PARAMETERS
C        (I*4)    KT   = NUMBER OF QUADRATURE POINTS - THRES. TYPE
C        (R*8)    XT() = FIXED QUADRATURE POINTS - THRES. TYPE
C        (R*8)    WT() = FIXED QUADRATURE POINTS - THRES. TYPE
C        (I*4)    K    = NUMBER OF QUADRATURE POINTS - ASYMP. TYPE
C        (R*8)    X()  = FIXED QUADRATURE POINTS - ASYMP. TYPE
C        (R*8)    W()  = FIXED QUADRATURE POINTS - ASYMP. TYPE
C
C  OUTPUT:
C        (R*8)    UPS9 - UPSILON
C
C
C  ROUTINES: NONE
C
C  DATE:     24/06/99 VERSION 1.1
C  AUTHOR:   HUGH SUMMERS, UNIVERSITY OF STRATHCLYDE
C - FIRST RELEASE
C
C-----------------------------------------------------------------------
      INTEGER             IT,          K,           KT
      REAL*8              B,           C,           C1,          C2
      REAL*8              E,           E0,          P(9),        T
      REAL*8              W(20),       WT(20),      X(20),       XT(20)
```

2.77 wupsilon: Subroutine wupsilon from library adas1xx

```
      SUBROUTINE WUPSILON( ELO   , EHI   , DELTAE, KTYPE , GF    ,
     &                     CPAR  , YKN   , I1    , I2    , NELEC ,
     &                     NCHAR , NTEMP , T     , UPS   , DSFULL,
     &                     ISTDIM, NET   , ENTE  , OMUP  , IWRITE,
     &                     INDIM , WI    , WJ
     &                   )
C-----------------------------------------------------------------------
C
C  ******************** FORTRAN77 SUBROUTINE:WUPSILON*****************
C
C  PURPOSE: TO WRITE DATA TO AN OLD/NEW ARCHIVE IN BURGESS FORMAT
C
C  INPUT:
C        (R*8) ELO    - LOWER LEVEL ENERGY
C        (R*8) EHI    - UPPER LEVEL ENERGY
C        (R*8) DELTAE - TRANSITION ENERGY
C        (I)   KTYPE  - TRANSITION TYPE
C        (R*8) GF     - WEIGHTED OSCILLATOR STRENGTH
C        (R*8) CPAR   - SCALABLE PARAMETER
C        (R*8) YKN    - KNOT POINTS
C        (I)   I1     - LOWER LEVEL INDEX
C        (I)   I2     - UPPER LEVEL INDEX
C        (R*8) NELEC  - NUMBER OF ELECTRONS
C        (R*8) NCHAR  - NUCLEAR CHARGE
C        (I)   NTEMP  - NUMBER OF TEMPERATURE POINTS
C        (R*8) T      - TEMPERATURES
C        (R*8) UPS    - UPSILONS
C        (C*80)DSFULL - ARCHIVE FILE NAME
C        (I)   ISTDIM - MAXIMUM INPUT ARRAY DIMENSIONS
C        (I)   NET    - NUMBER OF ENERGY POINTS
C        (R*8) ENTE   - ENRGIES
C        (R*8) OMUP   - OMEGAS
C        (I)   IWRITE - OUTPUT UNIT NUMBER
C        (I)   INDIM  - MAXIMUM OUTPUT ARRAY DIMENSION
C        (R*8) WI     - LOWER LEVEL STATISTICAL WEIGHT
C        (R*8) WJ     - UPPER LEVEL STATISTICAL WEIGHT
C
C  OUTPUT:
C
C  DATA:
C         CIARR(500)*80- 500 IS THE CURRENT LIMIT ON INDEXES
C  ROUTINES:
C         NONE
C  AUTHOR:
C         DAVID H.BROOKS (UNIV.OF STRATHCLYDE) EXT.4213/4205
C
C  DATE: 28/06/95 VERSION 1.1
C  UPDATED: DAVID H.BROOKS
C - FIRST RELEASE
C
C  DATE: 13/05/99 VERSION 1.2
C  MODIFIED: HUGH SUMMERS
C - CORRECTED CONFUSION ABOUT NCHAR AND NELEC IN COMMENTS
C                 INSERTED LEADING DECIMAL IN OUTPUT FLOATING FORMATS
C-----------------------------------------------------------------------
      CHARACTER*80        DSFULL
      INTEGER             I1,          I2,          INDIM,       ISTDIM
      INTEGER             IWRITE,      KTYPE,       NET,         NTEMP
      REAL*8              CPAR,        DELTAE,      EHI,         ELO
```

```
REAL*8              ENTE(ISTDIM),              GF,          NCHAR
REAL*8              NELEC,       OMUP(ISTDIM)
REAL*8              T(ISTDIM),   UPS(INDIM),   WI,          WJ
REAL*8              YKN(5)
```

2.78 wupsilon: Subroutine wupsilon from library adas1xx

```
      SUBROUTINE WUPSILON( ELO   , EHI   , DELTAE, KTYPE , GF    ,
     &                     CPAR  , YKN   , I1    , I2    , NELEC ,
     &                     NCHAR , NTEMP , T     , UPS   , DSFULL,
     &                     ISTDIM, NET   , ENTE  , OMUP  , IWRITE,
     &                     INDIM , WI    , WJ
     &                   )
C-----------------------------------------------------------------------
C
C  ******************** FORTRAN77 SUBROUTINE:WUPSILON*****************
C
C  PURPOSE: TO WRITE DATA TO AN OLD/NEW ARCHIVE IN BURGESS FORMAT
C
C  INPUT:
C        (R*8) ELO    - LOWER LEVEL ENERGY
C        (R*8) EHI    - UPPER LEVEL ENERGY
C        (R*8) DELTAE - TRANSITION ENERGY
C        (I)   KTYPE  - TRANSITION TYPE
C        (R*8) GF     - WEIGHTED OSCILLATOR STRENGTH
C        (R*8) CPAR   - SCALABLE PARAMETER
C        (R*8) YKN    - KNOT POINTS
C        (I)   I1     - LOWER LEVEL INDEX
C        (I)   I2     - UPPER LEVEL INDEX
C        (R*8) NELEC  - NUMBER OF ELECTRONS
C        (R*8) NCHAR  - NUCLEAR CHARGE
C        (I)   NTEMP  - NUMBER OF TEMPERATURE POINTS
C        (R*8) T      - TEMPERATURES
C        (R*8) UPS    - UPSILONS
C        (C*80)DSFULL - ARCHIVE FILE NAME
C        (I)   ISTDIM - MAXIMUM INPUT ARRAY DIMENSIONS
C        (I)   NET    - NUMBER OF ENERGY POINTS
C        (R*8) ENTE   - ENRGIES
C        (R*8) OMUP   - OMEGAS
C        (I)   IWRITE - OUTPUT UNIT NUMBER
C        (I)   INDIM  - MAXIMUM OUTPUT ARRAY DIMENSION
C        (R*8) WI     - LOWER LEVEL STATISTICAL WEIGHT
C        (R*8) WJ     - UPPER LEVEL STATISTICAL WEIGHT
C
C  OUTPUT:
C
C  DATA:
C         CIARR(500)*80- 500 IS THE CURRENT LIMIT ON INDEXES
C  ROUTINES:
C         NONE
C  AUTHOR:
C         DAVID H.BROOKS (UNIV.OF STRATHCLYDE) EXT.4213/4205
C
C  DATE: 28/06/95 VERSION 1.1
C  UPDATED: DAVID H.BROOKS
C - FIRST RELEASE
C
C  DATE: 13/05/99 VERSION 1.2
C  MODIFIED: HUGH SUMMERS
C - CORRECTED CONFUSION ABOUT NCHAR AND NELEC IN COMMENTS
C                 INSERTED LEADING DECIMAL IN OUTPUT FLOATING FORMATS
C-----------------------------------------------------------------------
      CHARACTER*80        DSFULL
      INTEGER             I1,          I2,          INDIM,       ISTDIM
      INTEGER             IWRITE,      KTYPE,       NET,         NTEMP
      REAL*8              CPAR,        DELTAE,      EHI,         ELO
```

```
REAL*8              ENTE(ISTDIM),              GF,          NCHAR
REAL*8              NELEC,       OMUP(ISTDIM)
REAL*8              T(ISTDIM),   UPS(INDIM),  WI,          WJ
REAL*8              YKN(5)
```

2.79 wupsilon9: Subroutine wupsilon9 from library adas1xx

```
      SUBROUTINE WUPSILON9( ELO   , EHI   , DELTAE, KTYPE , GF    ,
     &                      BPAR  , CPAR  , YKN   , I1    , I2    ,
     &                      NELEC , NCHAR , NTEMP , T     , UPS   ,
     &                      DSFULL, ISTDIM, NET   , ENTE  , OMUP  ,
     &                      IWRITE, INDIM , WI    , WJ
     &                    )
C-----------------------------------------------------------------------
C
C  ******************** FORTRAN77 SUBROUTINE:WUPSILON *****************
C
C  PURPOSE: TO WRITE DATA TO AN OLD/NEW ARCHIVE IN BURGESS FORMAT
C
C  INPUT:
C        (R*8) ELO    - LOWER LEVEL ENERGY
C        (R*8) EHI    - UPPER LEVEL ENERGY
C        (R*8) DELTAE - TRANSITION ENERGY
C        (I)   KTYPE  - TRANSITION TYPE
C        (R*8) GF     - WEIGHTED OSCILLATOR STRENGTH
C        (R*8) BPAR   - BURGESS SCALING PARAMETER - B
C        (R*8) CPAR   - BURGESS SCALING PARAMETER - C
C        (R*8) YKN    - KNOT POINTS
C        (I)   I1     - LOWER LEVEL INDEX
C        (I)   I2     - UPPER LEVEL INDEX
C        (R*8) NELEC  - NUMBER OF ELECTRONS
C        (R*8) NCHAR  - NUCLEAR CHARGE
C        (I)   NTEMP  - NUMBER OF TEMPERATURE POINTS
C        (R*8) T      - TEMPERATURES
C        (R*8) UPS    - UPSILONS
C        (C*80)DSFULL - ARCHIVE FILE NAME
C        (I)   ISTDIM - MAXIMUM INPUT ARRAY DIMENSIONS
C        (I)   NET    - NUMBER OF ENERGY POINTS
C        (R*8) ENTE   - ENRGIES
C        (R*8) OMUP   - OMEGAS
C        (I)   IWRITE - OUTPUT UNIT NUMBER
C        (I)   INDIM  - MAXIMUM OUTPUT ARRAY DIMENSION
C        (R*8) WI     - LOWER LEVEL STATISTICAL WEIGHT
C        (R*8) WJ     - UPPER LEVEL STATISTICAL WEIGHT
C
C  OUTPUT:
C
C  DATA:
C         CIARR(500)*80- 500 IS THE CURRENT LIMIT ON INDEXES
C  ROUTINES:
C         NONE
C
C  AUTHOR: HUGH SUMMERS, UNIVERSITY OF STRATHCLYDE
C  DATE: 18/06/99 VERSION 1.1
C
C-----------------------------------------------------------------------
      CHARACTER*80        DSFULL
      INTEGER             I1,          I2,          INDIM,       ISTDIM
      INTEGER             IWRITE,      KTYPE,       NET,         NTEMP
      REAL*8              BPAR,        CPAR,        DELTAE,      EHI
      REAL*8              ELO,         ENTE(ISTDIM),             GF
      REAL*8              NCHAR,       NELEC,       OMUP(ISTDIM)
      REAL*8              T(ISTDIM),   UPS(INDIM),  WI,          WJ
      REAL*8              YKN(9)
```

3 Subroutine library adas2xx

3.1 adwlpol: Subroutine adwlpol from library adas2xx

```
       SUBROUTINE ADWLPOL(Z0,NLQS,NSHELL,NA,LA,EA,QDA,ALFAA,JSN,JEALFA,
     &ACC,XMAX,H,LAM,IREPT,IEXT,ANS,OPEN17)
       IMPLICIT REAL*8(A-H,O-Z)
C-----------------------------------------------------------------------
C  VERSION OF DWLPOL FOR USE BY ADASRRC.  IT AVOIDS ALFA SEARCH FOR FREE
C  WAVE FUNCTIONS BY USING SAME SCREENING PARAMETERS AS BOUND STATE
C  ************** H.P. SUMMERS, JET   30 JUNE 1992 **********************
C
C  PURPOSE: EVALUATES LAM-POLE RADIAL MATRIX ELEMENTS USING DISTORTED
C  WAVES
C
C  BOUND-BOUND, BOUND-FREE AND FREE-FREE CASES ARE HANDLED.  THE
C  DISTORTED WAVES ARE IN A JUCYS OR SLATER TYPE POTENTIAL.
C  ************** H.P. SUMMERS, JET  24 APRIL 1985 **********************
C  INPUT
C      Z0=NUCLEAR CHARGE (+VE)
C      NLQS(I)=N,L,IQ FOR EACH SCREENING SHELL I=1 TO NSHELL
C      NSHELL=NUMBER OF SCREENING SHELLS
C      NA(1),NA(2)=INITIAL AND FINAL STATE PRINCIPAL QUANTUM NUMBERS.
C                  SET TO ZERO FOR FREE STATES
C      LA(1),LA(2)=INITIAL AND FINAL STATE ORBITAL QUANTUM NUMBERS.
C      EA(1),EA(2)=ENERGIES(RYD) OF INITIAL AND FINAL STATES
C                  SET <0 FOR BOUND STATES, SET >0 FOR FREE STATES.
C      QDA(1),QDA(2)=QUANTUM DEFECTS FOR INITIAL AND FINAL STATES.
C                    EXTRAPOLATED QUANTUM DEFECT USED FOR FREE STATE
C      ALFAA(1,I),ALFAA(2,I)=SCREENING PARAMETERS FOR INITIAL AND FINAL
C                          STATES FOR EACH SHELL I=1 TO NSHELL.
C      JSN=-1 JUCYS POTENTIAL
C         =0  SLATER POTENTIAL
C      JEALFA=0 SEARCH FOR ENERGIES GIVEN POTENTIAL (NO EFFECT FOR
C                  FREE STATES)
C            =1 SEARCH FOR ALFAA PARAMETERS FOR POTENTIAL GIVEN ENERGIES
C                  AND QUANTUM DEFECTS.
C      ACC=SEARCH ACCURACY SETTING
C      XMAX=RANGE FOR NUMERICAL WAVE FUNCTION GENERATION AND STORAGE
C      H=STEP INTERVAL FOR NUMERICAL WAVE FUNCTION STORAGE
C      LAM=MULTIPOLE (FOR RADIAL INTEGRAL <X**LAM>)
C      IREPT=0 FULL WAVE FUNCTION DETERMINATION
C           =1 REPETITION WITH SAME WAVE FUNCTIONS AS IN PREVIOUS CASE
C           =2 USE SAME BOUND WAVE FUNCTIONS AS IN PREVIOUS CASE,
C              USE FREE WAVE FUNCTIONS IN SAME POTENTIAL AS IN PREVIOUS
C              BUT WITH POSSIBLY DIFFERENT ENERGIES.
C      IEXT=0 NORMAL OPERATION WITH INTERNALLY GENERATED WAVE FUNCTIONS
C          =1 USE EXTERNAL WAVE FUNCTIONS SUPPLIED IN FUNCTION
C             GEXT(X,N,L) WITH N AND L SPECIFYING ORBITAL.
C      OPEN17 = FLAG WHETHER UNIT 17 IS OPENED OR NOT
C  OUTPUT
C      ANS=RADIAL INTEGRAL (AT. UNITS)
C
C
C  UPDATE:  HP SUMMERS  16/06/95  ALTER DEFINIAITON OF NLQS AS
C                                 1000*N+100*L+IQ TO AVOID PROBLEM WHEN\
C                                 NUMBER OF EQUIVALENT ELECTRONS IS 10.
C UNIX-IDL PORT:
C
C AUTHOR:  WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C
```

```
C DATE:    4TH JULY 1996
C
C VERSION: 1.1                          DATE: 04-07-96
C MODIFIED: WILLIAM OSBORN
C               - FIRST VERSION.
C
C VERSION: 1.2                          DATE: 19-08-96
C MODIFIED: WILLIAM OSBORN
C               - COMMENTED-OUT DIAGNOSTIC OUTPUT.
C               - ADDED OPEN17 PARAMETER.
C
C VERSION: 1.3                          DATE: 23-08-96
C MODIFIED: WILLIAM OSBORN
C               - CORRECTED OUTPUT TO STREAM 17
C
C VERSION: 1.4                          DATE: 19-12-01
C MODIFIED: Martin O'Mullane
C               - Removed junk from > column 72.
C
C VERSION: 1.5                          DATE: 16-05-07
C MODIFIED: Allan Whiteford
C               - Modified comments as part of subroutine documentation
C                 procedure.
C-----------------------------------------------------------------------
      INTEGER             IEXT,         IREPT,       JEALFA,      JSN
      INTEGER             LA(2),        LAM,         NA(2)
      INTEGER             NLQS(10),     NSHELL
      LOGICAL             OPEN17
      REAL*8              ACC,          ALFAA(2,10), ANS,         EA(2)
      REAL*8              H,            QDA(2),      XMAX,        Z0
```

3.2 adwrd2: Subroutine adwrd2 from library adas2xx

```
       FUNCTION ADWRD2(N,L,L1)
       IMPLICIT REAL*8(A-H,O-Z)
C -----------------------------------------------------------------------
C
C  VERSION OF DWRD2 FOR USE BY ADASRRC WHICH MAKES USE OF ADWLPOL
C  *********  H.P. SUMMERS, JET    30 JUNE 1992 ***********************
C
C  PURPOSE: CALCULATES SQUARE OF BOUND FREE DIPOLE INTEGRAL IN
C  DISTORTED WAVE APPROXIMATION.
C
C  THIS FUNCTION ACTS AS AN INTERFACE BETWEEN GIIDW AND DWLPOL.
C  ATOMIC STRUCTURE AND POTENTIAL DATA IS BROUGHT IN VIA LABELLED
C  COMMON BLOCK  /DWPARS/
C  *********  H.P. SUMMERS, JET 19 AUGUST 1985  ***********************
C  INPUT
C      N=PRINCIPAL QUANTUM NUMBER OF BOUND ELECTRON
C      L=ORBITAL QUANTUM NUMBER OF BOUND ELECTRON
C      L1=ORBITAL QUANTUM NUMBER OF FREE ELECTRON
C  OUTPUT
C      ADWRD2=SQUARED RADIAL DIPOLE INTEGRAL
C -----------------------------------------------------------------------
C-----------------------------------------------------------------------
C UNIX-IDL PORT:
C
C AUTHOR:  WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C
C DATE:    4TH JULY 1996
C
C VERSION: 1.1                          DATE: 04-07-96
C MODIFIED: WILLIAM OSBORN
C               - FIRST VERSION.
C VERSION: 1.2                          DATE: 16-05-07
C MODIFIED: Allan Whiteford
C               - Modified comments as part of subroutine documentation
C                 procedure.
C
C-----------------------------------------------------------------------
      INTEGER             L,           L1,          N
```

3.3 agiidw: Subroutine agiidw from library adas2xx

```
       FUNCTION AGIIDW(VVE,V,N,L,L1,LP,ISP,LT,LT1,IS,IRESOL)
       IMPLICIT REAL*8(A-H,O-Z)
C -----------------------------------------------------------------------
C
C  VERSION OF GIIDW FOR USE BY APHOTDW CALLED BY ADASRRC
C  ******************** H.P.SUMMERS, JET   30 JUNE 1992 ***************
C
C  PURPOSE: CALCULATES BOUND-FREE G-FACTORS USING DISTORTED WAVES,
C  BURGESS-SEATON PEACH OR HYDROGENIC APPROXIMATIONS
C
C  ******************** H.P.SUMMERS, JET   19 AUG. 1984 ***************
C  MAY SELECT DISTORTED WAVE MATRIX ELEMENTS, FROM PEACH TABLES, FROM
C  ORIGINAL BURGESS-SEATON APPROXIMATION OR HYDROGENIC MATRIX ELEMENTS
C  USING SELECTOR IBSOPT IN THE /BSPARS/ COMMON BLOCK.
C  FOR COMPLETENESS, THE UNRESOLVED, BUNDLED N, GBF (BURGESS AND SUMMERS
C  ,1976) CAN ALSO BE OBTAINED.
C  THE DRIVING PROGRAM MUST SET COMMON BLOCKS /PCHGTB/ AND /PCHXTB/ FOR
C  USE BY FUNCTIONS PCHG AND PCHX AND SET IFIRST=IGONE=1 AT START UP.
C  /PCHGTB/ DATA IS REQUIRED FROM FILE PCHGTAB.DATA ON STREAM 13
C  /PCHXTB/ DATA IS REQUIRED FROM FILE PCHXTAB.DATA ON STREAM 14
C  INPUT
C      VVE=V**2*E WHERE E=(FREE ELECTRON ENERGY)/Z**2 (RYD)
C      V=EFFECTIVE PRINCIPAL QUANTUM NUMBER OF BOUND ELECTRON
C      N=PRINCIPAL QUANTUM NUMBER OF BOUND ELECTRON
C      L=ORBITAL QUANTUM NUMBER OF BOUND ELECTRON
C      L1=ORBITAL QUANTUM NUMBER OF FREE ELECTRON
C      ISP=2*SP+1 WHERE SP IS TOTAL SPIN OF PARENT STATE
C      LP=TOTAL ORBITAL ANGULAR MOMENTUM QUANTUM NUMBER OF PARENT STATE
C      LT=TOTAL ORBITAL ANGULAR MOMENTUM QUANTUM NUMBER OF BOUND SYSTEM
C      LT1=TOTAL ORBITAL ANGULAR MOMENTUM QUANTUM NUMBER OF FREE SYSTEM
C      IS=2*S+1 WHERE S IS TOTAL SPIN OF SYSTEM
C      IRESOL=1 GIVES GII((LP,SP)N L LT S,(LP,SP)E L1 LT1 S)
C            =2 GIVES GII((LP,SP)N L LT S,(LP,SP)E L1 S) =ABOVE LT1 SUM
C            =3 GIVES GII((LP,SP)N L S,(LP,SP)E L1 S)   = ABOVE LT SUM
C            =4 GIVES GII((LP,SP)N L,(LP,SP)E L1)      = ABOVE S SUM
C            =5 GIVES GII(N,E) = GBF  (BURGESS AND SUMMERS)
C  OUTPUT
C      AGIIDW  THE BOUND-FREE GAUNT FACTOR
C-----------------------------------------------------------------------
C  UPDATE:  01/10/96  HP SUMMERS - BYPASS PEACH DATA INPUT IF IBSOPT=3
C-----------------------------------------------------------------------
C UNIX-IDL PORT:
C
C AUTHOR:  WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C
C DATE:    4TH JULY 1996
C
C VERSION: 1.1                          DATE: 04-07-96
C MODIFIED: WILLIAM OSBORN
C               - FIRST VERSION.
C VERSION: 1.2                          DATE: 14-10-96
C MODIFIED: WILLIAM OSBORN
C               - ADDED CHANGES DATED 01/10/96 ABOVE.
C
C VERSION: 1.3                          DATE: 16-05-07
C MODIFIED: Allan Whiteford
C               - Modified comments as part of subroutine documentation
C                 procedure.
C
```

```
C-----------------------------------------------------------------------
C -----------------------------------------------------------------------
      INTEGER             IRESOL,      IS,          ISP,         L
      INTEGER             L1,          LP,          LT,          LT1
      INTEGER             N
      REAL*8              V,           VVE
```

3.4 aphotdw: Subroutine aphotdw from library adas2xx

```
       SUBROUTINE APHOTDW(B,B1,V,N,L,L1,LP,ISP,LT,LT1,IS,PREC,PION,
      &PSTIM,IRESOL,ndgnt,gaunt,energy,ie)
       IMPLICIT REAL*8(A-H,O-Z)
C ----------------------------------------------------------------------
C
C  VERSION OF PHOTDW FOR USE BY ADASRRC WHICH USES AGIIDW
C  ******************* H.P.SUMMERS, JET   30 JUNE 1992 ***************
C
C  PURPOSE: CALCULATE PHOTO INTEGRALS USING GIIDW BOUND-FREE
C           GAUNT-FACTORS
C
C  SAME AS RECOM.FORT(PHOTO5) BUT CALLS GIIDW
C  ******************* H.P.SUMMERS, JET   19 AUG. 1984***************
C  INPUT
C      B=1.5789D5*Z**2/(V**2*TE)
C      B1=1.5789D5*Z**2/(V**2*TR)
C                WHERE
C                TE=ELECTRON TEMPERATURE (K)
C                TR=RADIATION TEMPERATURE (K)
C                Z=BOUND STATE ION CHARGE +1
C                (THUS Z**2/V***2 IS THE IONISATION POTENTIAL  (RYD))
C      V=EFFECTIVE PRINCIPAL QUANTUM NUMBER OF BOUND ELECTRON
C      N=PRINCIPAL QUANTUM NUMBER OF BOUND ELECTRON
C      L=ORBITAL QUANTUM NUMBER OF BOUND ELECTRON
C      L1=ORBITAL QUANTUM NUMBER OF FREE ELECTRON
C      ISP=2*SP+1 WHERE SP IS TOTAL SPIN OF PARENT STATE
C      LP=TOTAL ORBITAL ANGULAR MOMENTUM QUANTUM NUMBER OF PARENT STATE
C      LT=TOTAL ORBITAL ANGULAR MOMENTUM QUANTUM NUMBER OF BOUND SYSTEM
C      LT1=TOTAL ORBITAL ANGULAR MOMENTUM QUANTUM NUMBER OF FREE SYSTEM
C      IS=2*S+1 WHERE S IS TOTAL SPIN OF SYSTEM
C      ndgnt = max number of Gaunt factors allowed
C  OUTPUT
C      PREC=RADIATIVE RECOMBINATION INTEGRAL
C      PION=PHOTOIONISATION INTEGRAL
C      PSTIM=STIMULATED RECOMBINATION INTEGRAL
C          WHERE
C      IRESOL=1 FOR  ((LP,SP)N L LT S,(LP,SP)L1 LT1 S)
C             =2 FOR  ((LP,SP)N L LT S,(LP,SP)L1 S) =ABOVE LT1 SUM
C             =3 FOR  ((LP,SP)N L S,(LP,SP)L1 S)  = ABOVE LT SUM
C             =4 FOR  ((LP,SP)N L,(LP,SP)L1)  = ABOVE S SUM
C             =5 FOR  NO L RESOLUTION USING GBF
C      gaunt()  = Bound-free Gaunt factor at energy
C      energy() = v**2*e
C                 where e = (free electron energy)/z**2 (ryd)
C                       v = effective principal quantum number
C                           of bound electron
C      ie       = number of Gaunt/energy pairs
C
C
C UNIX-IDL PORT:
C
C AUTHOR:  WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C
C DATE:    4TH JULY 1996
C
C VERSION: 1.1                          DATE: 04-07-96
C MODIFIED: WILLIAM OSBORN
C               - FIRST VERSION.
C
```

```
C VERSION: 1.2                          DATE: 19-12-01
C MODIFIED: Martin O'MULLANE
C               - Removed junk from > column 72.
C
C VERSION: 1.3                          DATE: 02-02-05
C MODIFIED: Paul Bryans
C               - Returns Gaunt factor, associated vve and number
C                 of Gaunt/vve pairs
C
C VERSION: 1.4                          DATE: 16-05-07
C MODIFIED: Allan Whiteford
C               - Modified comments as part of subroutine documentation
C                 procedure.
C
CC ------------------------------------------------------------------------
C
      INTEGER             IE,           IRESOL,      IS,           ISP
      INTEGER             L,            L1,          LP,           LT
      INTEGER             LT1,          N,           NDGNT
      REAL*8              B,            B1,          ENERGY(NDGNT)
      REAL*8              GAUNT(NDGNT),              PION,         PREC
      REAL*8              PSTIM,        V
```

3.5 ass: Subroutine ass from library adas2xx

```
C
       subroutine ass(a10,a1,a20,a2,phi1,phi2,x,n,e1,e2,nmax,rem)
c-----------------------------------------------------------------------
c
c  ***************** fortran77 subroutine: ass   *********************
c
c  purpose:  calculates asymptotic integrals using power series modules
c
c
c  subroutine:
c
c  input : (r*8)  a10
c  input : (r*8)  a1
c  input : (r*8)  a20
c  input : (r*8)  a2
c  input : (r*8)  phi1
c  input : (r*8)  phi2
c  input : (r*8)  x
c  input : (i*4)  n
c  input : (r*8)  e1
c  input : (r*8)  e2
c  input : (i*4)  nmax
c
c  output: (r*8)  rem
c
c
c  routines:
c          routine    source    brief description
c          ------------------------------------------------------------
c          dnaq       adas
c          dnprod     adas
c          i4unit     adas      fetch unit number for output of messages
c
c  author:  h. p. summers, university of strathclyde
c           ja7.08
c           tel. 0141-548-4196
c
c  date:    06/06/02
c
c  update:

C VERSION: 1.1                          DATE: 04-07-96
C MODIFIED: WILLIAM OSBORN
C               - FIRST VERSION.
C VERSION: 1.2                          DATE: 19-12-01
C MODIFIED: Martin O'MULLANE
C               - Removed junk from > column 72.
C
C VERSION: 1.3                          DATE: 18-03-03
C MODIFIED: Hugh Summers
C               - re-written and documented
c
c-----------------------------------------------------------------------
c-----------------------------------------------------------------------
      INTEGER             N,          NMAX
      REAL*8              A1(20),     A10,         A2(20),      A20
      REAL*8              E1,         E2,          PHI1,        PHI2
      REAL*8              REM,        X
```

3.6 ass2: Subroutine ass2 from library adas2xx

```
C
       subroutine ass2( x1    , h     , x    , f0    , f1    ,
     &                  g0    , g1    , ei   , ej    , tkij  ,
     &                  li    , lj    , z    , ni    , rem
     &                )
c-----------------------------------------------------------------------
c
c  ****************** fortran77 subroutine: ass2   *********************
c
c  purpose:  calculate asymptotic part of integrals over wave functions
c
c
c  subroutine:
c
c  input : (r*8)  x1
c  input : (r*8)  h
c  input : (r*8)  x
c  input : (r*8)  f0
c  input : (r*8)  f1
c  input : (r*8)  g0
c  input : (r*8)  g1
c  input : (r*8)  ei
c  input : (r*8)  ej
c  input : (r*8)  tkij
c  input : (i*4)  li
c  input : (i*4)  lj
c  input : (r*8)  z
c  input : (i*4)  ni
c
c  output: (r*8)  rem
c
c  routines:
c          none
c
c  author:  h. p. summers, university of strathclyde
c           ja7.08
c           tel. 0141-548-4196
c
c  date:    06/06/02
c
c  update:
C
C VERSION: 1.1                          DATE: 04-07-96
C MODIFIED: WILLIAM OSBORN
C               - FIRST VERSION.
C
C VERSION: 1.2                          DATE: 19-12-01
C MODIFIED: Martin O'MULLANE
C               - Removed junk from > column 72.c
C
C VERSION: 1.3                          DATE: 18-03-03
C MODIFIED: Hugh Summers
C               -Re-written and documented.
C
c-----------------------------------------------------------------------
c-----------------------------------------------------------------------
      INTEGER             LI,          LJ,          NI
      REAL*8              EI,          EJ,          F0,          F1
      REAL*8              G0,          G1,          H,           REM
```

```
      REAL*8              TKIJ,        X,           X1,          Z
```

3.7 b1data: Subroutine b1data from library adas2xx

```
       SUBROUTINE B1DATA( IUNIT  , NDLEV  , NDTRN ,
     &                    TITLED , IZ     , IZ0   , IZ1   , BWNO  ,
     &                    IL     ,
     &                    IA     , CSTRGA , ISA   , ILA   , XJA   , WA ,
     &                    NV     , SCEF   ,
     &                    ITRAN  ,
     &                    I1A    , I2A    , AVAL  , SCOM
     &                  )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: B1DATA *********************
C
C  PURPOSE:  TO FETCH DATA FROM INPUT SPECIFIC Z EXCITATION FILE.
C            (ELECTRON IMPACT TRANSITIONS ONLY).
C
C  CALLING PROGRAM: ADAS201
C
C  DATA:
C           THE 'REAL' DATA IN THE FILE IS REPRESENTED IN AN ABBREVIATED
C           FORM WHICH OMITS THE "D" OR "E" EXPONENT SPECIFIER.
C           e.g. 1.23D-06 or 1.23E-06 IS REPRESENTED AS 1.23-06
C                6.75D+07 or 6.75E+07 IS REPRESENTED AS 6.75+07
C
C           THEREFORE THE FORM OF EACH 'REAL' NUMBER IN THE DATA SET IS:
C                          N.NN+NN or N.NN-NN
C
C           THE UNITS USED IN THE DATA FILE ARE TAKEN AS FOLLOWS:
C
C           IONISATION POTENTIAL: WAVE NUMBER (CM-1)
C           INDEX LEVEL ENERGIES: WAVE NUMBER (CM-1)
C           TEMPERATURES        : KELVIN
C           A-VALUES            : SEC-1
C           GAMMA-VALUES        :
C           RATE COEFFT.        : CM3 SEC-1
C
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  IUNIT   = UNIT TO WHICH INPUT FILE IS ALLOCATED
C  INPUT : (I*4)  NDLEV   = MAXIMUM NUMBER OF LEVELS THAT CAN BE READ
C  INPUT : (I*4)  NDTRN   = MAX. NUMBER OF TRANSITIONS THAT CAN BE READ
C
C  OUTPUT: (C*3)  TITLED  = ELEMENT SYMBOL.
C  OUTPUT: (I*4)  IZ      =  RECOMBINED ION CHARGE READ
C  OUTPUT: (I*4)  IZ0     =         NUCLEAR CHARGE READ
C  OUTPUT: (I*4)  IZ1     = RECOMBINING ION CHARGE READ
C                           (NOTE: IZ1 SHOULD EQUAL IZ+1)
C  OUTPUT: (R*8)  BWNO    = IONISATION POTENTIAL (CM-1)
C
C  OUTPUT: (I*4)  IL      = INPUT DATA FILE: NUMBER OF ENERGY LEVELS
C
C  OUTPUT: (I*4)  IA()    = ENERGY LEVEL INDEX NUMBER
C  OUTPUT: (C*12) CSTRGA()= NOMENCLATURE/CONFIGURATION FOR LEVEL 'IA()'
C  OUTPUT: (I*4)  ISA()   = MULTIPLICITY FOR LEVEL 'IA()'
C                           NOTE: (ISA-1)/2 = QUANTUM NUMBER (S)
C  OUTPUT: (I*4)  ILA()   = QUANTUM NUMBER (L) FOR LEVEL 'IA()'
C  OUTPUT: (R*8)  XJA()   = QUANTUM NUMBER (J-VALUE) FOR LEVEL 'IA()'
C                           NOTE: (2*XJA)+1 = STATISTICAL WEIGHT
C  OUTPUT: (R*8)  WA()    = ENERGY RELATIVE TO LEVEL 1 (CM-1) FOR LEVEL
```

```
C                          'IA()'
C
C  OUTPUT: (I*4)  NV      = INPUT DATA FILE: NUMBER OF GAMMA/TEMPERATURE
C                            PAIRS FOR A GIVEN TRANSITION.
C  OUTPUT: (R*8)  SCEF()  = INPUT DATA FILE: ELECTRON TEMPERATURES (K)
C                            (INITIALLY JUST THE MANTISSA. SEE 'ITPOW()')
C                            (NOTE: TE=TP=TH IS ASSUMED)
C
C  OUTPUT: (I*4)  ITRAN   = INPUT DATA FILE: NUMBER OF ELECTRON IMPACT
C                                            TRANSITIONS.
C
C  OUTPUT: (I*4)  I1A()   = ELECTRON IMPACT TRANSITION:
C                             LOWER ENERGY LEVEL INDEX
C  OUTPUT: (I*4)  I2A()   = ELECTRON IMPACT TRANSITION:
C                             UPPER ENERGY LEVEL INDEX
C  OUTPUT: (R*8)  AVAL()  = ELECTRON IMPACT TRANSITION:
C                             A-VALUE (SEC-1)
C  OUTPUT: (R*8)  SCOM(,) = ELECTRON IMPACT TRANSITION:
C                             GAMMA VALUES
C                            1ST DIMENSION - TEMPERATURE 'SCEF()'
C                            2ND DIMENSION - TRANSITION NUMBER
C
C          (I*4)  NDTEM   = PARAMETER = MAX NUMBER OF INPUT FILE TEMPS.
C
C          (I*4)  I4UNIT  = FUNCTION (SEE ROUTINE SECTION BELOW)
C          (I*4)  IQS     = X-SECT DATA FORMAT SELECTOR
C                            NOTE: IQS=3 ONLY ALLOWED IN THIS PROGRAM
C          (I*4)  I       = GENERAL USE.
C          (I*4)  J       = GENERAL USE.
C          (I*4)  J1      = INPUT DATA FILE - SELECTED TRANSITION:
C                             LOWER ENERGY LEVEL INDEX (CASE ' ' & 'P')
C          (I*4)  J2      = INPUT DATA FILE - SELECTED TRANSITION:
C                             UPPER ENERGY LEVEL INDEX (CASE ' ' & 'P')
C                             CAPTURING    LEVEL INDEX (CASE 'H' & 'R')
C          (I*4)  ILINE   = ENERGY LEVEL INDEX FOR CURRENT LINE
C          (I*4)  IAPOW   = EXPONENT OF 'AVALM'
C          (I*4)  IGPOW() = EXPONENT OF 'GAMMA()'
C          (I*4)  ITPOW() = TEMPERATURES - EXPONENT
C                            NOTE: MANTISSA INITIALLY KEPT IN 'SCEF()'
C
C          (R*4)  ZF      = SHOULD BE EQUIVALENT TO 'IZ1'
C
C          (R*8)  AVALM   = INPUT DATA FILE - SELECTED TRANSITION:
C                            MANTISSA OF:   ('IAPOW' => EXPONENT)
C                             A-VALUE (SEC-1)          (CASE ' ')
C                             NEUTRAL BEAM ENERGY      (CASE 'H')
C                             NOT USED                 (CASE 'P' & 'R')
C          (R*8)  GAMMA() = INPUT DATA FILE - SELECTED TRANSITION:
C                            MANTISSA OF: ('IGPOW()' => EXPONENT)
C                             GAMMA VALUES             (CASE ' ' & 'P')
C                             RATE COEFFT. (CM3 SEC-1) (CASE 'H' & 'R')
C                            DIMENSION => TEMPERATURE 'SCEF()'
C
C          (C*1)  TCODE   = TRANSITION: DATA TYPE POINTER:
C                            ' ' => Electron Impact   Transition
C                            'P' => Proton   Impact   Transition
C                            'H' => Charge   Exchange Recombination
C                            'R' => Free     Electron Recombination
C          (C*80) CLINE   = CURRENT ENERGY LEVEL INDEX PARAMETER LINE
C
C          (L*4)  LDATA   = IDENTIFIES  WHETHER  THE END OF AN  INPUT
```

```
C                           SECTION IN THE DATA SET HAS BEEN LOCATED.
C                           (.TRUE. => END OF SECTION REACHED)
C
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4UNIT     ADAS      FETCH UNIT NUMBER FOR OUTPUT OF MESSAGES
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 4569
C
C DATE:    09/10/90
C
C UPDATE:  16/11/90 - LEVEL LINE READ AS A CHARACTER*80 STRING FIRST
C                     (PE BRIDEN)
C
C UPDATE:  23/04/93 - PE BRIDEN - ADAS91: ADDED I4UNIT FUNCTION TO WRITE
C                                         STATEMENTS FOR SCREEN MESSAGES
C
C UPDATE:  24/05/93 - PE BRIDEN - ADAS91: CHANGED I4UNIT(0)-> I4UNIT(-1)
C
C-----------------------------------------------------------------------
      CHARACTER*12        CSTRGA(NDLEV)
      CHARACTER*3         TITLED
      INTEGER             I1A(NDTRN),  I2A(NDTRN),  IA(NDLEV),   IL
      INTEGER             ILA(NDLEV),  ISA(NDLEV),  ITRAN,       IUNIT
      INTEGER             IZ,          IZ0,         IZ1,         NDLEV
      INTEGER             NDTRN,       NV
      REAL*8              AVAL(NDTRN), BWNO,        SCEF(NDTEM)
      REAL*8              SCOM(NDTEM,NDTRN),        WA(NDLEV)
      REAL*8              XJA(NDLEV)
```

3.8 b1rate: Subroutine b1rate from library adas2xx

```
      SUBROUTINE B1RATE( NARR  , TEMP   , GAMMA  ,
     &                           EUPPER , ELOWER ,
     &                           WUPPER , WLOWER ,
     &                           RATE   , DRATE
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: B1RATE *********************
C
C  PURPOSE: TO CALCULATE THE EXCITATION AND DE-EXCIATATION RATE COEFFI-
C           CIENTS FOR A SET OF INPUT TEMPERATURE(kelvin)/ GAMMA PAIRS.
C
C  CALLING PROGRAM:  ADAS201
C
C  SUBROUTINE:
C
C  INPUT :  (I*4)  NARR    = NUMBER OF INPUT TEMPERATURE/GAMMA PAIRS
C  INPUT :  (R*8)  TEMP()  = TEMPERATURE VALUES (kelvin)
C  INPUT :  (R*8)  GAMMA() = GAMMA VALUES
C
C  INPUT :  (R*8)  EUPPER  = SELECTED TRANSITION - UPPER ENERGY LEVEL
C                            RELATIVE TO INDEX LEVEL 1 (CM-1).
C  INPUT :  (R*8)  ELOWER  = SELECTED TRANSITION - LOWER ENERGY LEVEL
C                            RELATIVE TO INDEX LEVEL 1 (CM-1).
C
C  INPUT :  (R*8)  WUPPER  = SELECTED TRANSITION - UPPER ENERGY LEVEL
C                            STATISTICAL WEIGHT.
C  INPUT :  (R*8)  WLOWER  = SELECTED TRANSITION - LOWER ENERGY LEVEL
C                            STATISTICAL WEIGHT.
C
C  OUTPUT:  (R*8)  RATE    = EXCITATION RATE COEFFS (cm**3/s)
C  OUTPUT:  (R*8)  DRATE   = DEEXCITATION RATE COEFS (cm**3/s)
C
C           (R*8)  TK2ATE  = PARAMETER = EQUATION CONSTANT = 1.5789D+05
C           (R*8)  R2GAM   = PARAMETER = EQUATION CONSTANT = 2.17161D-08
C           (R*8)  WN2RYD  = PARAMETER =
C                            WAVE NUMBER (CM-1) TO RYDBERG CONVERSION
C
C           (I*4)  I       = GENERAL ARRAY INDEX
C
C           (R*8)  SUPPER  = 1/(UPPER LEVEL STATISTICAL WEIGHT)
C           (R*8)  SLOWER  = 1/(LOWER LEVEL STATISTICAL WEIGHT)
C           (R*8)  RYDDIF  = NEGATIVE TRANSITION ENERGY IN RYDBERGS
C                            ( NOTE: 1 Rydberg = 1.09737E5 cm-1)
C           (R*8)  ATE     = EQUATION PARAMETER
C           (R*8)  GVAL    = EQUATION PARAMETER
C
C ROUTINES:  NONE
C
C NOTES:
C            EQUATIONS USED -
C
C                      2.17161E-8 x GAMMA x SQRT(157890 / TEMP)
C            RATE = ------------------------------------------
C                   WLOWER x EXP(1.4388 x (EUPPER-ELOWER) / TEMP)
C
C                      2.17161E-8 x GAMMA x SQRT(157890 / TEMP)
C            DRATE = ------------------------------------------
C                                  WUPPER
```

```
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 4569
C
C DATE:    09/10/90
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             NARR
      REAL*8              DRATE(NARR), ELOWER,      EUPPER
      REAL*8              GAMMA(NARR), RATE(NARR),  TEMP(NARR),  WLOWER
      REAL*8              WUPPER
```

3.9 b1spln: Subroutine b1spln from library adas2xx

```
      SUBROUTINE B1SPLN(         LOSEL ,
     &                   NV    , MAXT  , NPSPL  ,
     &                   SCEF  , TOA   , TOSA   ,
     &                   GAMMA , GAMOA , GAMOSA ,
     &                   LTRNG
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: B1SPLN *********************
C
C  (IDENTICAL TO: C1SPLN (EXCEPT SOME VARIABLE NAMES ARE CHANGED))
C
C  PURPOSE:
C         1) PERFORMS CUBIC SPLINE ON LOG(TEMPERATURE) VERSUS LOG(GAMMA)
C            INPUT DATA. ('SCEF' VERSUS 'GAMMA' , NV DATA PAIRS)
C
C         2) INTERPOLATES 'MAXT'  GAMMA VALUES USING ABOVE SPLINES AT
C            TEMPERATURES READ IN FROM ISPF PANELS FOR TABULAR OUTPUT.
C            (ANY TEMPERATURE VALUES WHICH REQUIRED EXTRAPOLATION TO
C             TAKE PLACE ARE SET TO ZERO).
C                 - THIS STEP ONLY TAKES PLACE IF 'LOSEL=.TRUE.' -
C
C         3) INTERPOLATES 'NPSPL' GAMMA VALUES USING ABOVE SPLINES AT
C            TEMPERATURES EQUI-DISTANCE ON RANGE OF LOG(TEMPERATURES)
C            STORED IN INPUT 'SCEF' ARRAY.
C
C  CALLING PROGRAM: ADAS201
C
C
C  SUBROUTINE:
C
C  INPUT : (L*4)  LOSEL   = .TRUE.  => CALCULATE GAMMAS FOR INPUT TEMPS.
C                                      READ FROM ISPF PANEL.
C                           .FALSE. => - DO NOT DO THE ABOVE -
C
C  INPUT : (I*4)  NV      = INPUT DATA FILE: NUMBER OF GAMMA/TEMPERATURE
C                           PAIRS READ FOR THE TRANSITION BEING ASSESSED
C  INPUT : (I*4)  MAXT    = NUMBER OF ISPF ENTERED TEMPERATURE VALUES AT
C                           WHICH INTERPOLATED GAMMA VALUES ARE REQUIRED
C                           FOR TABULAR OUTPUT.
C  INPUT : (I*4)  NPSPL   = NUMBER OF  SPLINE  INTERPOLATED  GAMMA/TEMP.
C                           REQUIRED FOR GRAPHICAL DISPLAY.
C
C  INPUT : (I*4)  SCEF()  = INPUT DATA FILE: TEMPERATURES (KELVIN)
C  INPUT : (I*4)  TOA()   = ISPF PANEL ENTERED TEMPERATURES (KELVIN)
C  OUTPUT: (I*4)  TOSA()  = 'NPSPL' TEMPERATURES FOR GRAPHICAL OUTPUT
C                           (KELVIN).
C
C  INPUT : (R*8)  GAMMA() = INPUT DATA FILE: SELECTED TRANSITION -
C                           GAMMA VALUES AT 'SCEF()'.
C  OUTPUT: (I*4)  GAMOA() = SPLINE INTERPOLATED GAMMA VALUES AT 'TOA()'
C                           (EXTRAPOLATED VALUES = 0.0).
C  OUTPUT: (R*8)  GAMOSA()= SPLINE INTERPOLATED GAMMA VALUES AT 'TOSA()'
C
C  OUTPUT: (L*4)  LTRNG() = .TRUE.  => OUTPUT   SPLINE    VALUE    WAS
C                                      INTERPOLATED FOR 'DLOG(TOA()'.
C                           .FALSE. => OUTPUT   SPLINE    VALUE    WAS
C                                      EXTRAPOLATED FOR 'DLOG(TOA()'.
```

```
C                                     (NOTE: 'YOUT()=0' AS 'IOPT < 0').
C
C          (I*4)  NIN     = PARAMETER = MAX. NO. OF  INPUT  TEMP/GAMMA
C                                       PAIRS MUST BE >= 'NV'
C          (I*4)  NOUT    = PARAMETER = MAX. NO. OF 'OUTPUT TEMP/GAMMA
C                                       PAIRS MUST BE >= 'MAXT' & 'NPSPL'
C
C          (I*4)  IARR    = ARRAY SUBSCRIPT USED FOR TEMP/GAMMA PAIRS
C          (I*4)  IOPT    = DEFINES THE BOUNDARY DERIVATIVES FOR THE
C                           SPLINE ROUTINE 'XXSPLE', SEE 'XXSPLE'.
C                           (VALID VALUES = <0, 0, 1, 2, 3, 4)
C
C          (R*8)  TSTEP   = THE SIZE OF STEP BETWEEN 'XOUT()' VALUES FOR
C                           GRAPHICAL  OUTPUT  TEMP/GAMMA  PAIRS  TO  BE
C                           CALCULATED USING SPLINES.
C
C          (L*4)  LSETX   = .TRUE.  => SET UP SPLINE PARAMETERS RELATING
C                                      TO 'XIN' AXIS.
C                           .FALSE. => DO NOT SET UP SPLINE PARAMETERS
C                                      RELATING TO 'XIN' AXIS.
C                                      (I.E. THEY WERE SET IN A PREVIOUS
C                                            CALL )
C                           (VALUE SET TO .FALSE. BY 'XXSPLE')
C
C          (R*8)  XIN()   = LOG( 'SCEF()' )
C          (R*8)  YIN()   = LOG( 'GAMMA()' )
C          (R*8)  XOUT()  = LOG(TEMPERATURES AT WHICH SPLINES REQUIRED)
C          (R*8)  YOUT()  = LOG(OUTPUT SPLINE INTERPOLATED GAMMA VALUES)
C          (R*8)  DF()    = SPLINE INTERPOLATED DERIVATIVES
C
C          (L*4)  LDUMP() = .TRUE.  => OUTPUT SPLINE VALUE INTRPOLATED
C                                      FOR 'YOUT()'.
C                           .FALSE. => OUTPUT SPLINE VALUE EXTRAPOLATED
C                                      FOR 'YOUT()'.
C                                      (NOTE: USED AS A DUMMY ARGUMENT.
C                                             ALL VALUES WILL BE TRUE.)
C
C NOTE:
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          XXSPLE     ADAS      SPLINE SUBROUTINE (EXTENDED DIAGNOSTICS)
C          R8FUN1     ADAS      REAL*8 FUNCTION: ( X -> X )
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 4569
C
C DATE:    05/02/91
C
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C VERSION:  1.2 DATE: 27-09-95
C - INCREASED PARAMETER NIN FROM 8 TO 14 (SAME
C   CHANGE MADE IN THE JET IBM CODES BY PAUL BRIDEN).
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             MAXT,        NPSPL,       NV
      LOGICAL             LOSEL,       LTRNG(MAXT)
```

```
REAL*8              GAMMA(NV),    GAMOA(MAXT), GAMOSA(NPSPL)
REAL*8              SCEF(NV),     TOA(MAXT),   TOSA(NPSPL)
```

3.10 b1tran: Subroutine b1tran from library adas2xx

```
       SUBROUTINE B1TRAN( NDLEV  , NDTRN  , NDTEM ,
     &                    IL     , ISTRN  , NV    ,
     &                    IA     , WA     , XJA   ,
     &                    I1A    , I2A    , AVAL  , SCOM ,
     &                    IUPPER , ILOWER ,
     &                    LUPPER , LLOWER ,
     &                    WUPPER , WLOWER ,
     &                    EUPPER , ELOWER ,
     &                    AA     , GAMMA
     &                  )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: B1TRAN *********************
C
C  PURPOSE:  TO SET UP SELECTED TRANSITION PARAMETERS.
C
C  CALLING PROGRAM: ADAS201
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  NDLEV   = MAXIMUM NUMBER OF INDEX LEVELS
C  INPUT : (I*4)  NDTRN   = MAXIMUM NUMBER OF TRANSITIONS
C  INPUT : (I*4)  NDTEM   = MAXIMUM NUMBER OF INPUT FILE TEMPERATURES
C
C  INPUT : (I*4)  IL      = NUMBER OF INDEX LEVELS
C  INPUT : (I*4)  ISTRN   = SELECTED TRANSITION INDEX.
C  INPUT : (I*4)  NV      = INPUT DATA FILE: NUMBER OF GAMMA/TEMPERATURE
C                           PAIRS FOR THE SELECTED TRANSITION.
C
C  INPUT : (I*4)  IA()    = LEVEL INDEX NUMBER ARRAY
C  INPUT : (R*8)  WA()    = LEVEL ENERGIES RELATIVE TO LEVEL 1 (CM-1)
C  INPUT : (R*8)  XJA()   = QUANTUM NUMBER (J-VALUE) FOR LEVEL
C                           NOTE: (2*XJA)+1 = STATISTICAL WEIGHT
C
C  INPUT : (I*4)  I1A()   = LOWER LEVEL INDEX FOR ELECTRON IMPACT
C                           TRANSITION
C  INPUT : (I*4)  I2A()   = UPPER LEVEL INDEX FOR ELECTRON IMPACT
C                           TRANSITION
C  INPUT : (I*4)  AVAL()  = A-VALUE FOR ELECTRON IMPACT TRANSITION
C  INPUT : (I*4)  SCOM(,) = GAMMA VALUES FOR ELECTRON IMPACT TRANSITION
C                            1st DIMENSION: TEMPERATURE INDEX
C                            2nd DIMENSION: TRANSITION INDEX
C
C  OUTPUT: (I*4)  IUPPER  = SELECTED TRANSITION: UPPER LEVEL ARRAY INDEX
C  OUTPUT: (I*4)  ILOWER  = SELECTED TRANSITION: LOWER LEVEL ARRAY INDEX
C
C
C  OUTPUT: (I*4)  LUPPER  = SELECTED TRANSITION: UPPER INDEX LEVEL
C  OUTPUT: (I*4)  LLOWER  = SELECTED TRANSITION: LOWER INDEX LEVEL
C
C  OUTPUT: (R*8)  WUPPER  = SELECTED TRANSITION: UPPER LEVEL STAT. WT.
C  OUTPUT: (R*8)  WLOWER  = SELECTED TRANSITION: LOWER LEVEL STAR. WT.
C                           (NOTE: STAT. WT. = STATISTICAL WEIGHT)
C
C  OUTPUT: (R*8)  EUPPER  = SELECTED TRANSITION: UPPER ENERGY LEVEL
C                           RELATIVE TO INDEX LEVEL 1. (CM-1)
C  OUTPUT: (R*8)  ELOWER  = SELECTED TRANSITION: LOWER ENERGY LEVEL
C                           RELATIVE TO INDEX LEVEL 1. (CM-1)
C
```

```
C  OUTPUT: (R*8)  GAMMA() = INPUT DATA FILE: SELECTED TRANSITION -
C                                           GAMMA VALUE AT 'TEMP()'
C  OUTPUT: (R*8)  AA      = SELECTED TRANSITION A-VALUE (SEC-1)
C
C          (I*4)  I       = GENERAL USE.
C
C ROUTINES: NONE
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 4569
C
C DATE:    09/10/90
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      INTEGER             I1A(NDTRN),  I2A(NDTRN),  IA(NDLEV),   IL
      INTEGER             ILOWER,      ISTRN,       IUPPER,      LLOWER
      INTEGER             LUPPER,      NDLEV,       NDTEM,       NDTRN
      INTEGER             NV
      REAL*8              AA,          AVAL(NDTRN), ELOWER,      EUPPER
      REAL*8              GAMMA(NDTEM),             SCOM(NDTEM,NDTRN)
      REAL*8              WA(NDLEV),   WLOWER,      WUPPER
      REAL*8              XJA(NDLEV)
```

3.11 b2gspc: Subroutine b2gspc from library adas2xx

```
       SUBROUTINE B2GSPC(XA,N,C1,C2,C3,C4)
C
       IMPLICIT REAL*8(A-H,O-Z)
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE : B2GSPC ********************
C
C PURPOSE:
C  SUBROUTINE TO GENERATE PRECURSORS OF SPLINE COEFFICIENTS SUITABLE
C  FOR BOTH FORWARD AND BACKWARD INTERPOLATION
C
C
C
C  INPUT
C      XA(I)=SET OF KNOTS
C      N=NUMBER OF KNOTS  (N.LE.20)
C  OUTPUT
C      C1(I,J)=1ST SPLINE COEFFICIENT PRECURSOR
C      C2(I,J)=2ND SPLINE COEFFICIENT PRECURSOR
C      C3(I,J)=3RD SPLINE COEFFICIENT PRECURSOR
C      C4(I,J)=4TH SPLINE COEFFICIENT PRECURSOR
C
C NOTES: THIS ROUTINE IS NOT YET PROPERLY ANNOTATED
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 06-03-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - PUT UNDER S.C.C.S. CONTROL
C
C-----------------------------------------------------------------------
C
      INTEGER             N
      REAL*8              C1(10,9),    C2(10,9),    C3(10,9)
      REAL*8              C4(10,9),    XA(10)
```

3.12 b2nfas: Subroutine b2nfas from library adas2xx

```
       SUBROUTINE B2NFAS(X,XA,N,YA,Y,DY,C1,C2,C3,C4,FORM,IFORMS)
C
       IMPLICIT REAL*8(A-H,O-Z)
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: B2NFAS *********************
C
C PURPOSE:
C  SUBROUTINE TO PROVIDE A SPLINE INTERPOLATE MAKING USE OF SPECIFIED
C  ASYMPTOTIC BEHAVIOUR
C
C
C  USES LABELLED COMMON /SPL3/
C
C  INPUT
C      X=REQUIRED X-VALUE
C      X(I)=KNOTS
C      N=NUMBER OF KNOTS
C      C1(I,J)=1ST SPINE COEFFICIENT PRECURSOR
C      C2(I,J)=2ND SPINE COEFFICIENT PRECURSOR
C      C3(I,J)=3RD SPINE COEFFICIENT PRECURSOR
C      C4(I,J)=4TH SPINE COEFFICIENT PRECURSOR
C      FORM=EXTERNAL FUNCTION SPECIFYING ASYMPTOTIC FORMS
C      IFORMS=INDEX OF REQUIRED FORM
C  OUTPUT
C      Y=RETURNED Y-VALUE
C      DY=RETURNED DERIVATIVE
C
C NOTES: THIS ROUTINE IS NOT YET PROPERLY ANNOTATED
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 06-03-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - PUT UNDER S.C.C.S. CONTROL
C
C-----------------------------------------------------------------------
C
      INTEGER             IFORMS,      N
      REAL*8              C1(10,9),    C2(10,9),    C3(10,9)
      REAL*8              C4(10,9),    DY,          X,           XA(10)
      REAL*8              Y,           YA(10)
```

3.13 b2nfit: Subroutine b2nfit from library adas2xx

```
       SUBROUTINE B2NFIT(X,XA,N,YAA,Y,DY,I0,C1,C2,C3,C4,ISW)
C
       IMPLICIT REAL*8(A-H,O-Z)
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: B2NFIT *********************
C
C PURPOSE:
C  SUBROUTINE TO PERFORM SPLINE INTERPOLATION
C
C
C  INPUT
C      X       = REQUIRED X-VALUE
C      XA(I)   = X-VALUES
C      N       = NUMBER OF VALUES
C      YAA(I)  = Y-VALUES (POSSIBLY STORED AS MULTIPLE SETS)
C      I0      = STARTING INDEX(-1) IN YAA ARRAY OF REQUIRED INPUT SET
C      C1(I,J) = 1ST SPLINE COEFFICIENT PRECURSOR
C      C2(I,J) = 2ND SPLINE COEFFICIENT PRECURSOR
C      C3(I,J) = 3RD SPLINE COEFFICIENT PRECURSOR
C      C4(I,J) = 4TH SPLINE COEFFICIENT PRECURSOR
C      ISW     = .LE.0  ORDINARY     SPLINE INTERPOLATION
C              = .GT.0  LOGARITHMIC  SPLINE INTERPOLATION
C  OUTPUT
C      Y       = RETURNED Y-VALUE
C      DY      = RETURNED DERIVATIVE
C
C NOTES: THIS ROUTINE IS NOT YET PROPERLY ANNOTATED
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 06-03-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - PUT UNDER S.C.C.S. CONTROL
C
C-----------------------------------------------------------------------
C
      INTEGER             I0,           ISW,          N
      REAL*8              C1(10,9),     C2(10,9),     C3(10,9)
      REAL*8              C4(10,9),     DY,           X,           XA(10)
      REAL*8              Y,            YAA(10)
```

3.14 b2ngas: Subroutine b2ngas from library adas2xx

```
       SUBROUTINE B2NGAS(X,DX,FORM,IFORMS,IENDS)
C
       IMPLICIT REAL*8(A-H,O-Z)
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: B2NGAS *********************
C
C PURPOSE:
C  SUBROUTINE INITIALISES COMMON ARRAYS REQUIRED FOR SPLINING WITH
C  SMOOTH FITTING TO AN ASYMPTOTIC FORM
C
C
C  USES LABELLED COMMON /SPL3/
C
C  IF IENDS=1,MATCHING IS AT FIRST KNOT(GIVEN BY X)
C          =2,MATCHING IS AT  LAST KNOT(GIVEN BY X)
C  ASYMPTOTIC FORMS ARE GIVEN IN THE EXTERNAL FUNCTION FORM(I,X)
C  WHERE I=4*IFORMS-5+2*IENDS POINTS TO FIRST PART OF ASYMP. FORM
C         =4*IFORMS-4+2*IENDS POINTS TO SECOND PART OF ASYMP. FORM
C  THUS A SERIES OF ASYMPTOTIC FORMS MAY BE PRESENT IN FORM
C
C  INPUT
C      COMMON /SPL3/ PROVIDES INPUT IN VECTOR IEND WHICH SPECIFIES
C                    CHOICE OF END CONDITION AT FIRST IEND(1) OR LAST
C                    IEND(2) KNOT OF SPLINE
C      X=X-VALUE OF END POINT
C      DX=DISPLACEMENT FROM X-VALUE FOR DERIVATIVE EVALUATION
C      FORM=EXTERNAL FUNCTION SPECIFYING ASYMPTOTIC FORMS
C      IFORMS=SELECTED FORM
C      IENDS=1,MATCHING IS AT FIRST KNOT(GIVEN BY X)
C           =2,MATCHING IS AT  LAST KNOT(GIVEN BY X)
C  OUTPUT
C      COMMON /SPL3/ IS SET BY THIS ROUTINE
C
C NOTES: THIS ROUTINE IS NOT YET PROPERLY ANNOTATED
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 06-03-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - PUT UNDER S.C.C.S. CONTROL
C
C-----------------------------------------------------------------------
C
       COMMON /SPL3/IEND(2),G(2),AB(4),PQ(12),ABRY(40)
       IF(IENDS.EQ.1.AND.IEND(1).EQ.4)GO TO 5
       IF(IENDS.EQ.2.AND.IEND(2).EQ.4)GO TO 5
    3  RETURN
    5  I=4*IFORMS-5+2*IENDS
       J=6*IENDS-5
       IC=0
       DX1=1.0D0/DX
   10  PQ(J)=FORM(I,X)
       T1=FORM(I,X+DX)
       T2=FORM(I,X-DX)
       PQ(J+1)=0.5D0*DX1*(T1-T2)
       PQ(J+2)=DX1*DX1*(T1-2.0D0*PQ(J)+T2)
       IC=IC+1
       IF(IC.GT.1)RETURN
```

```
 I=I+1
 J=J+3
 GO TO 10
END
INTEGER             IENDS,       IFORMS
REAL*8              DX,          X
```

3.15 b2sort: Subroutine b2sort from library adas2xx

```
       SUBROUTINE B2SORT(XA,YA,N)
C
       IMPLICIT REAL*8(A-H,O-Z)
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: B2SORT *********************
C
C PURPOSE : TO SORT AN ARRAY SO THAT XA IS IN INCREASING ORDER
C
C  N.B.  INPUT VALUES ARE ALTERED BY THIS ROUTINE !!!!
C
C  INPUT
C      XA(I)=X-VALUES
C      YA(I)=Y-VALUES
C      N=NUMBER OF VALUES
C  OUTPUT
C      XA(I)=SORTED X-VALUES
C      YA(I)=SORTED Y-VALUES
C
C NOTES: THIS ROUTINE IS NOT YET PROPERLY ANNOTATED
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 06-03-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - PUT UNDER S.C.C.S. CONTROL
C
C-----------------------------------------------------------------------
C
      INTEGER             N
      REAL*8              XA(10),      YA(10)
```

3.16 b2spij3: Subroutine b2spij3 from library adas2xx

```
       SUBROUTINE B2SPIJ3(N,H,W)
C
       IMPLICIT REAL*8(A-H,O-Z)
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: B2SPIJ3 *******************
C
C PURPOSE:
C  SUBROUTINE TO CALCULATE SPLINES WITH VARIOUS END CONDITIONS.
C
C
C  USES LABELLED COMMON  /SPL3/
C
C  CONDITIONS AT 1ST NODE AND NTH NODE CONTROLLED BY IEND1 AND IENDN
C      IEND=1 : SPECIFIED D LOG(Y) IE. DY/Y AT NODE STORED IN
C               APPROPRIATE VECTOR
C          =2 : ZERO CURVATURE
C          =3 : CONSTANT CURVATURE
C          =4 : MATCHED TO SPECIFIED FUNCTIONAL FORM IN TERMS OF
C               TWO PARAMETERS A AND B SUCH THAT
C                        FUNCT = P(1)*A+Q(1)*B
C                    1ST DERIV. = P(2)*A+Q(2)*B
C                    2ND DERIV. = P(3)*A+Q(3)*B
C               WHERE A1,B1,P1,Q1 ARE USED FOR 1ST NODE AND
C               AN,BN,PN,QN FOR NTH NODE
C
C  INPUT
C      N=NUMBER OF KNOTS
C      H(I)=INTERVALS BETWEEN KNOTS
C  OUTPUT
C      W=SPLINE MATRIX
C
C NOTES: THIS ROUTINE IS NOT YET PROPERLY ANNOTATED
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 06-03-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - PUT UNDER S.C.C.S. CONTROL
C
C-----------------------------------------------------------------------
C
      INTEGER            N
      REAL*8             H(10),       W(10,10)
```

3.17 b3data: Subroutine b3data from library adas2xx

```
       SUBROUTINE B3DATA( IUNIT , NDLEV , NDTRN , NDTEM , IZDIMD,
     &                    SEQSYM, IZMAX , Z1A   , IZA   , IZ0A  , IZ1A,
     &                    BWNOA , IL    ,
     &                    IA    , CSTRGA, NA    , ISA   , ILA   , XJA  ,
     &                    WAA   , ITRAN ,
     &                    NVA   , SCEFA ,
     &                    I1A   , I2A   , N1A   , N2A   , W1A   , W2A,
     &                    IEC1A , IAC1A , IAC2A , FAC2A , IGC1A , FGC2A,
     &                    CTSTRA, WDEA  , AVALA , SCOMA , LADJA
     &                  )
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: B3DATA *********************
C
C  PURPOSE:  TO FETCH DATA FROM INPUT GENERAL Z EXCITATION FILE.
C            (ELECTRON IMPACT TRANSITIONS ONLY).
C
C  CALLING PROGRAM: ADAS203
C
C  DATA:
C
C           THE UNITS USED IN THE DATA FILE ARE TAKEN AS FOLLOWS:
C
C           IONISATION POTENTIAL: WAVE NUMBER (CM-1)
C           INDEX LEVEL ENERGIES: WAVE NUMBER (CM-1)
C           TEMPERATURES        : KELVIN
C           A-VALUES            : SEC-1
C           GAMMA-VALUES        :
C
C
C  SUBROUTINE:
C
C          (I*4)  NDZ     = PARAMETER = MAXIMUM NUMBER OF IONS IN
C                                       A GENERAL Z FILE
C
C  INPUT : (I*4)  IUNIT   = UNIT TO WHICH INPUT FILE IS ALLOCATED
C  INPUT : (I*4)  NDLEV   = MAXIMUM NUMBER OF LEVELS THAT CAN BE READ
C  INPUT : (I*4)  NDTRN   = MAX. NUMBER OF TRANSITIONS THAT CAN BE READ
C  INPUT : (I*4)  NDTEM   = MAX NUMBER OF INPUT FILE TEMPS.
C  INPUT : (I*4)  IZDIMD  = MAX. NUMBER OF SEQUENCE MEMBERS ALLOWED
C
C  OUTPUT: (C*2)  SEQSYM  = ELEMENT SYMBOL.
C  OUTPUT: (I*4)  IZMAX   = NUMBER OF SEQUENCE MEMBERS
C  OUTPUT: (R*8)  Z1A()   = SEQUNCE RECOMBINING ION CHARGES READ
C                           1ST DIMENSION - SEQUENCE MEMBER INDEX
C  OUTPUT: (I*4)  IZA()   = SEQUENCE RECOMBINED ION CHARGES
C                           1ST DIMENSION - SEQUENCE MEMBER INDEX
C  OUTPUT: (I*4)  IZ0A()  = SEQUENCE NUCLEAR CHARGES
C                           1ST DIMENSION - SEQUENCE MEMBER INDEX
C  OUTPUT: (I*4)  IZ1A()  = SEQUNCE RECOMBINING ION CHARGES READ
C                           1ST DIMENSION - SEQUENCE MEMBER INDEX
C                           (NOTE: IZ1 SHOULD EQUAL IZ+1)
C  OUTPUT: (R*8)  BWNOA() = IONISATION POTENTIALS (CM-1)
C                           1ST DIMENSION - SEQUENCE MEMBER INDEX
C
C  OUTPUT: (I*4)  IL      = INPUT DATA FILE: NUMBER OF ENERGY LEVELS
C
C  OUTPUT: (I*4)  IA()    = ENERGY LEVEL INDEX NUMBER
```

```
C  OUTPUT: (C*18) CSTRGA()= NOMENCLATURE/CONFIGURATION FOR LEVEL 'IA()'
C  OUTPUT: (I*4)  NA()    = PRINCIPAL QUANTUM NUMBER OF VALENCE ELECTRON
C  OUTPUT: (I*4)  ISA()   = MULTIPLICITY FOR LEVEL 'IA()'
C                           NOTE: (ISA-1)/2 = QUANTUM NUMBER (S)
C  OUTPUT: (I*4)  ILA()   = QUANTUM NUMBER (L) FOR LEVEL 'IA()'
C  OUTPUT: (R*8)  XJA()   = QUANTUM NUMBER (J-VALUE) FOR LEVEL 'IA()'
C                           NOTE: (2*XJA)+1 = STATISTICAL WEIGHT
C  OUTPUT: (R*8)  WAA()   = ENERGY RELATIVE TO LEVEL 1 (CM-1) FOR LEVEL
C                           'IA()'
C                           1ST DIMENSION - LEVEL INDEX
C                           2ND DIMENSION - SEQUENCE MEMBER INDEX
C  OUTPUT: (I*4)  ITRAN   = INPUT DATA FILE: NUMBER OF ELECTRON IMPACT
C                                            TRANSITIONS.
C
C  OUTPUT: (I*4)  NVA()   = INPUT DATA FILE: NUMBER OF GAMMA/TEMPERATURE
C                           PAIRS FOR A GIVEN TRANSITION.
C                           1ST DIMENSION - SEQUENCE MEMBER INDEX
C  OUTPUT: (R*8)  SCEFA(,)= INPUT DATA FILE: Z-SCALED ELEC. TEMPS.(K)
C                           1ST DIMENSION - TEMPERATURE 'SCEF()'
C                           2ND DIMENSION - TRANSITION NUMBER
C
C
C  OUTPUT: (I*4)  I1A()   = ELECTRON IMPACT TRANSITION:
C                            LOWER ENERGY LEVEL INDEX
C                           1ST DIMENSION - TRANSITION NUMBER
C  OUTPUT: (I*4)  I2A()   = ELECTRON IMPACT TRANSITION:
C                            UPPER ENERGY LEVEL INDEX
C                           1ST DIMENSION - TRANSITION NUMBER
C  OUTPUT: (I*4)  N1A()   = ELECTRON IMPACT TRANSITION:
C                            LOWER LEVEL PRINCIPAL QUANTUM NUMBER
C                           1ST DIMENSION - TRANSITION NUMBER
C  OUTPUT: (I*4)  N2A()   = ELECTRON IMPACT TRANSITION:
C                            UPPER LEVEL PRINCIPAL QUANTUM NUMBER
C                           1ST DIMENSION - TRANSITION NUMBER
C  OUTPUT: (I*4)  W1A()   = ELECTRON IMPACT TRANSITION:
C                            LOWER LEVEL STATISTICAL WEIGHT
C                           1ST DIMENSION - TRANSITION NUMBER
C  OUTPUT: (I*4)  W2A()   = ELECTRON IMPACT TRANSITION:
C                            UPPER LEVEL STATISTICAL WEIGHT
C                           1ST DIMENSION - TRANSITION NUMBER
C
C  OUTPUT: (I*4)  IEC1A() = TRANSITION ENERGY INTERPOLATION VARIABLE
C                           (1=>Z1 ; 2=>1/Z1)
C                           1ST DIMENSION - TRANSITION NUMBER
C  OUTPUT: (I*4)  IAC1A() = TRANSITION PROB. INTERPOLATION VARIABLE
C                           (1=>Z1 ; 2=> 1/Z1)
C                           1ST DIMENSION - TRANSITION NUMBER
C  OUTPUT: (I*4)  IAC2A() = TRANSITION TYPE
C                           (1=>DIPOLE, 2=>NON-DIPOLE, 3=>SPIN CHANGE,
C                            4=>OTHER)
C                           1ST DIMENSION - TRANSITION NUMBER
C  OUTPUT: (I*4)  FAC2A() = TRANSITION PROB. Z1 SCALING POWER
C                           1ST DIMENSION - TRANSITION NUMBER
C  OUTPUT: (I*4)  IGC1A() = UPSILON INTERPOLATION VARIABLE
C                           (1=>Z1 ; 2=> 1/Z1)
C                           1ST DIMENSION - TRANSITION NUMBER
C  OUTPUT: (I*4)  FGC2A() = UPSILON Z1 SCALING POWER
C                           1ST DIMENSION - TRANSITION NUMBER
C  OUTPUT: (C*18) CTSTRA()= TRANSITION DESCRIPTOR
C                           1ST DIMENSION - TRANSITION NUMBER
C  OUTPUT: (R*8)  WDEA()  = TRANSITION ENERGY (CM-1)
```

```
C                           1ST DIMENSION - TRANSITION NUMBER
C
C  OUTPUT: (R*8)  AVALA()  = ELECTRON IMPACT TRANSITION:
C                             A-VALUES (SEC-1)
C                           1ST DIMENSION - TRANSITION NUMBER
C                           2ND DIMENSION - SEQUENCE MEMBER INDEX
C  OUTPUT: (R*8)  SCOMA(,,)=ELECTRON IMPACT TRANSITION:
C                            GAMMA VALUES
C                           1ST DIMENSION - TEMPERATURE 'SCEF()'
C                           2ND DIMENSION - TRANSITION NUMBER
C                           3RD DIMENSION - SEQUENCE MEMBER INDEX
C  OUTPUT: (L*4)  LADJA()  = .FALSE. INITIALISATION NO TRANSITIONS
C                                    ADJUSTED YET
C                           1ST DIMENSION - TRANSITION NUMBER
C
C
C          (I*4)  I       = GENERAL USE.
C          (I*4)  IAC1    = CURRENT VALUE OF IAC1 PARAMETER
C          (I*4)  IAC2    = CURRENT VALUE OF IAC2 PARAMETER
C          (I*4)  IEC1    = CURRENT VALUE OF IEC1 PARAMETER
C          (I*4)  IGC1    = CURRENT VALUE OF IGC1 PARAMETER
C          (I*4)  ILINE   = ENERGY LEVEL INDEX FOR CURRENT LINE
C          (I*4)  ITEMP   = GENERAL USE.
C          (I*4)  IZ      = CURRENT ION CHARGE
C          (I*4)  IZS     = ISOELECTRONIC SEQUENCE CHARGE (1ST MEMBER)
C          (I*4)  I4EIZ0  = FUNCTION (SEE ROUTINE SECTION BELOW)
C          (I*4)  I4UNIT  = FUNCTION (SEE ROUTINE SECTION BELOW)
C          (I*4)  J       = GENERAL USE.
C          (I*4)  J1      = INPUT DATA FILE - SELECTED TRANSITION:
C                           LOWER ENERGY LEVEL INDEX (CASE ' ' & 'P')
C          (I*4)  J2      = INPUT DATA FILE - SELECTED TRANSITION:
C                           UPPER ENERGY LEVEL INDEX (CASE ' ' & 'P')
C                           CAPTURING    LEVEL INDEX (CASE 'H' & 'R')
C          (I*4)  IPOS    = GENERAL POSITION IN STRING MARKER.
C          (I*4)  K       = GENERAL USE.
C          (I*4)  NI      = CURRENT LOWER LEVEL PRINCIPAL QUANTUM NUMBER
C          (I*4)  NJ      = CURRENT UPPER LEVEL PRINCIPAL QUANTUM NUMBER
C
C          (R*8)  FAC2    = CURRENT VALUE OF FAC2 PARAMETER
C          (R*8)  FGC2    = CURRENT VALUE OF FGC2 PARAMETER
C          (R*8)  WI      = CURRENT LOWER LEVEL STATISTICAL WEIGHT
C          (R*8)  WJ      = CURRENT UPPER LEVEL STATISTICAL WEIGHT
C
C          (C*2)  CHEQSYM = EXTRA STRING USED FOR SYMBOL CHECKING
C          (C*1)  CMINUS  = STRING OF MINUS SYMBOL USED TO CHECK FOR
C                           SINGLE CHARACTER ELEMENT SYMBOLS
C          (C*80) CLINE   = CURRENT ENERGY LEVEL INDEX PARAMETER LINE
C          (C*18) STRING  = GENERAL USE
C
C          (L*4)  LDATA   = IDENTIFIES  WHETHER  THE END OF AN  INPUT
C                           SECTION IN THE DATA SET HAS BEEN LOCATED.
C                           (.TRUE. => END OF SECTION REACHED)
C
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4EIZ0     ADAS      RETURN NUCLEAR CHARGE OF ELEMENT SYMBOL
C          I4UNIT     ADAS      FETCH UNIT NUMBER FOR OUTPUT OF MESSAGES
C
C AUTHOR:  H. P. SUMMERS, JET
```

```
C          K1/1/57
C          JET EXT. 4941
C
C DATE:    17/08/94
C
C UPDATE:  03/07/95 - HPS  ALTER LENGTH OF CSTRGA ARRAY TO 18 AND
C                          USE NEW POSITION FOR INPUT/OUTPUT
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 29-02-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - PUT UNDER S.C.C.S. CONTROL
C
C VERSION: 1.2                          DATE: 29-02-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - CORRECTED ERROR MESSAGES
C               - ADDED ABILITY TO HANDLE SINGLE LETTER ELEMENT
C                 SYMBOLS FOLLOWED BY DASH. E.G. IN HYDROGEN LIKE
C                 FILES THE FIRST TWO CHARACTERS IN THE FILE ARE 'H-'
C                 AND THIS WAS CAUSING PROBLEMS FOR I4EIZ0 BEFORE.
C
C-----------------------------------------------------------------------
      CHARACTER*18        CSTRGA(NDLEV),            CTSTRA(NDTRN)
      CHARACTER*2         SEQSYM
      INTEGER             I1A(NDTRN),  I2A(NDTRN),  IA(NDLEV)
      INTEGER             IAC1A(NDTRN),             IAC2A(NDTRN)
      INTEGER             IEC1A(NDTRN),             IGC1A(NDTRN)
      INTEGER             IL,          ILA(NDLEV),  ISA(NDLEV),  ITRAN
      INTEGER             IUNIT,       IZ0A(IZDIMD)
      INTEGER             IZ1A(IZDIMD),             IZA(IZDIMD), IZDIMD
      INTEGER             IZMAX,       N1A(NDTRN),  N2A(NDTRN)
      INTEGER             NA(NDLEV),   NDLEV,       NDTEM,       NDTRN
      INTEGER             NVA(NDTRN)
      LOGICAL             LADJA(NDTRN)
      REAL*8              AVALA(NDTRN,IZDIMD),      BWNOA(IZDIMD)
      REAL*8              FAC2A(NDTRN),             FGC2A(NDTRN)
      REAL*8              SCEFA(NDTEM,NDTRN)
      REAL*8              SCOMA(NDTEM,NDTRN,IZDIMD)
      REAL*8              W1A(NDTRN),  W2A(NDTRN),  WAA(NDLEV,IZDIMD)
      REAL*8              WDEA(NDTRN,IZDIMD),       XJA(NDLEV)
      REAL*8              Z1A(IZDIMD)
```

3.18 b3leve: Subroutine b3leve from library adas2xx

```
       SUBROUTINE B3LEVE( NDLEV , IZDIMD,
     &                    IZMAX , Z1A   , IZA   , IZ0A  , IZ1A,
     &                    BWNOA , IL    ,
     &                    IA    , NA    , WAA   ,
     &                    IZS   , IZ0   ,
     &                    BWNO  , WAO )
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: B3LEVE *********************
C
C  PURPOSE:  TO EVALUATE IONISATION AND LEVEL ENERGIES FOR A SELECTED
C            MEMBER OF AN ISOELECTRONIC SEQUENCE FROM THE GENERAL Z DATA
C
C  CALLING PROGRAM: ADAS203
C
C  DATA:
C
C           THE UNITS USED IN THE DATA FILE ARE TAKEN AS FOLLOWS:
C
C           IONISATION POTENTIAL: WAVE NUMBER (CM-1)
C           INDEX LEVEL ENERGIES: WAVE NUMBER (CM-1)
C
C
C  SUBROUTINE:
C
C          (I*4)  NDSPLN  = PARAMETER = MAXIMUM NUMBER OF SPLINE KNOTS
C
C  INPUT : (I*4)  NDLEV   = MAXIMUM NUMBER OF LEVELS THAT CAN BE READ
C  INPUT : (I*4)  IZDIMD  = MAX. NUMBER OF SEQUENCE MEMBERS ALLOWED
C
C  INPUT : (I*4)  IZMAX   = NUMBER OF SEQUENCE MEMBERS
C  INPUT : (R*8)  Z1A()   = SEQUNCE RECOMBINING ION CHARGES READ
C                           1ST DIMENSION - SEQUENCE MEMBER INDEX
C  INPUT : (I*4)  IZA()   = SEQUENCE RECOMBINED ION CHARGES
C                           1ST DIMENSION - SEQUENCE MEMBER INDEX
C  INPUT : (I*4)  IZ0A()  = SEQUENCE NUCLEAR CHARGES
C                           1ST DIMENSION - SEQUENCE MEMBER INDEX
C  INPUT : (I*4)  IZ1A()  = SEQUNCE RECOMBINING ION CHARGES READ
C                           1ST DIMENSION - SEQUENCE MEMBER INDEX
C                           (NOTE: IZ1 SHOULD EQUAL IZ+1)
C  INPUT : (R*8)  BWNOA() = IONISATION POTENTIALS (CM-1)
C                           1ST DIMENSION - SEQUENCE MEMBER INDEX
C
C  INPUT : (I*4)  IL      = INPUT DATA FILE: NUMBER OF ENERGY LEVELS
C
C  INPUT : (I*4)  IA()    = ENERGY LEVEL INDEX NUMBER
C  INPUT : (I*4)  NA()    = PRINCIPAL QUANTUM NUMBER OF VALENCE ELECTRON
C  INPUT : (R*8)  WAA()   = ENERGY RELATIVE TO LEVEL 1 (CM-1) FOR LEVEL
C                           'IA()'
C                           1ST DIMENSION - LEVEL INDEX
C                           2ND DIMENSION - SEQUENCE MEMBER INDEX
C  INPUT : (I*4)  IZS     = NUCLEAR CHARGE OF NEUTRAL SEQUENCE MEMBER
C  INPUT : (I*4)  IZ0     = NUCLEAR CHARGE OF SELECTED ION
C
C  OUTPUT: (R*8)  BWNO    = IONISATION ENERGY OF SELECTED ION (CM-1)
C  OUTPUT: (R*8)  WAO()   = LEVEL ENERGIES RELATIVE TO LOWEST (CM-1)
C
C          (I*4)  I       = GENERAL USE.
```

```
C          (I*4)  IENDN   = SPLINE END CONDITION SWITCH AT LAST POINT
C          (I*4)  IEND1   = SPLINE END CONDITION SWITCH AT FIRST POINT
C          (I*4)  IFORMS  = SPLINE INDEPENDENT VARIABLE FORM SWITCH
C          (I*4)  K       = GENERAL USE.
C
C          (R*8)  C1(,)   = 1ST SPLINE COEFFICIENT MATRIX
C          (R*8)  C2(,)   = 2ND SPLINE COEFFICIENT MATRIX
C          (R*8)  C3(,)   = 3RD SPLINE COEFFICIENT MATRIX
C          (R*8)  C4(,)   = 4TH SPLINE COEFFICIENT MATRIX
C          (R*8)  DY      = GRADIENT OF SPLINE AT POINT
C          (R*8)  ENI     = LEVEL PRINCIPAL QUANTUM NUMBER
C          (R*8)  EN1     = LOWEST LEVEL PRINCIPAL QUANTUM NUMBER
C          (R*8)  E1I     = LEVEL ENERGY (RYDBERGS)
C          (R*8)  B3FORM  = EXTERNAL FUNCTION (SEE SUBROUTINE SECTION)
C          (R*8)  REN     = GENERAL USE
C          (R*8)  XI      = GENERAL USE
C          (R*8)  XSA()   = SPLINE INDEPENDENT VARIABLE AT KNOTS
C          (R*8)  Y       = SPLINE INTERPOLATED VALUE
C          (R*8)  YSA()   = SPLINE DEPENDENT VARIABLE AT KNOTS
C          (R*8)  Z1      = CURRENT ION CHARGE +1
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          B2GSPC     ADAS      GENERATES SPLINE COEFFICIENT MATRICES
C          B2NFAS     ADAS      SETS SPLINE ASYMPTOTIC CONDITIONS
C          B3FORM     ADAS      INDEPENDENT VARIABLE FUNCTION FOR SPLINE
C          B2SORT     ADAS      SORTS VECTOR INTO INCREASING ORDER
C
C AUTHOR:  H. P. SUMMERS, JET
C          K1/1/57
C          JET EXT. 4941
C
C DATE:    08/01/95
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 20-03-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - PUT UNDER S.C.C.S. CONTROL
C               - REPLACED CALLS TO NSORT ROUTINE WITH CALLS TO B2SORT.
C                 NSORT IS USED TO SORT A REAL ARRAY AND ASSOCIATED
C                 INTEGER ARRAY WHEREAS WHAT WAS BEING PASSED TO IT WAS
C                 A REAL ARRAY AND ANOTHER, ASSOCIATED REAL ARRAY. B2SORT
C                 TAKES 2 REAL ARRAYS AS INPUT AND PERFORMS A BUBBLE SORT
C                 ON THEM.
C
C VERSION: 1.2                          DATE: 23-04-07
C MODIFIED: ALLAN WHITEFORD
C               - RENAMED FORM SUBROUTINE TO B3FORM.
C
C-----------------------------------------------------------------------
      INTEGER             IA(NDLEV),   IL,          IZ0
      INTEGER             IZ0A(IZDIMD),             IZ1A(IZDIMD)
      INTEGER             IZA(IZDIMD), IZDIMD,      IZMAX,       IZS
      INTEGER             NA(NDLEV),   NDLEV
      REAL*8              BWNO,        BWNOA(IZDIMD)
      REAL*8              WAA(NDLEV,IZDIMD),        WAO(NDLEV)
      REAL*8              Z1A(IZDIMD)
```

3.19 b3reac: Subroutine b3reac from library adas2xx

```
       SUBROUTINE B3REAC( NDTRN , NDTEM , IZDIMD,
     &                    IZMAX , Z1A   ,
     &                    ITRAN ,
     &                    NVA   , SCEFA ,
     &                    I1A   , I2A   , N1A   , N2A   , W1A   , W2A,
     &                    IEC1A , IAC1A , IAC2A , FAC2A , IGC1A , FGC2A,
     &                    CTSTRA, WDEA  , AVALA , SCOMA ,
     &                    IZS   , IZ0   , ISTRN , LIBPT ,
     &                    MAXT  , TEA   ,
     &                    WDE   , AVAL  , SCOM
     &                  )
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: B3REAC *********************
C
C  PURPOSE:  TO INTERPOLATE DATA FOR A SINGLE TRANSITION FROM A
C            GENERAL Z EXCITATION FILE TO A SELECTED SEQUENCE MEMBER
C
C  CALLING PROGRAM: ADAS203
C
C  DATA:
C
C           THE UNITS USED IN THE DATA FILE ARE TAKEN AS FOLLOWS:
C
C           IONISATION POTENTIAL: WAVE NUMBER (CM-1)
C           INDEX LEVEL ENERGIES: WAVE NUMBER (CM-1)
C           TEMPERATURES        : KELVIN
C           A-VALUES            : SEC-1
C           GAMMA-VALUES        :
C
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  NDTRN   = MAX. NUMBER OF TRANSITIONS THAT CAN BE READ
C  INPUT : (I*4)  NDTEM   = MAX NUMBER OF INPUT FILE TEMPS.
C  INPUT : (I*4)  IZDIMD  = MAX. NUMBER OF SEQUENCE MEMBERS ALLOWED
C
C  INPUT : (I*4)  IZMAX   = NO. OF SEQUENCE MEMBERS IN GENERAL Z FILE
C  INPUT : (R*8)  Z1A()   = ION CHARGE +1  FOR SEQUENCE MEBERS IN
C                           GENERAL Z FILE
C                           1ST DIMENSION - SEQUENCE MEMBER INDEX
C
C  INPUT : (I*4)  ITRAN   = INPUT DATA FILE: NUMBER OF ELECTRON IMPACT
C
C
C  INPUT : (I*4)  NVA()   = INPUT DATA FILE: NUMBER OF GAMMA/TEMPERATURE
C                           PAIRS FOR A GIVEN TRANSITION.
C                           1ST DIMENSION - SEQUENCE MEMBER INDEX
C  INPUT : (R*8)  SCEFA(,)= INPUT DATA FILE: Z-SCALED ELEC. TEMPS.(K)
C                           1ST DIMENSION - TEMPERATURE 'SCEF()'
C                           2ND DIMENSION - TRANSITION NUMBER
C                                           TRANSITIONS.
C
C
C  INPUT : (I*4)  I1A()   = ELECTRON IMPACT TRANSITION:
C                            LOWER ENERGY LEVEL INDEX
C                           1ST DIMENSION - TRANSITION NUMBER
C  INPUT : (I*4)  I2A()   = ELECTRON IMPACT TRANSITION:
```

```
C                              UPPER ENERGY LEVEL INDEX
C                             1ST DIMENSION - TRANSITION NUMBER
C  INPUT : (I*4)  N1A()    = ELECTRON IMPACT TRANSITION:
C                              LOWER LEVEL PRINCIPAL QUANTUM NUMBER
C                             1ST DIMENSION - TRANSITION NUMBER
C  INPUT : (I*4)  N2A()    = ELECTRON IMPACT TRANSITION:
C                              UPPER LEVEL PRINCIPAL QUANTUM NUMBER
C                             1ST DIMENSION - TRANSITION NUMBER
C  INPUT : (I*4)  W1A()    = ELECTRON IMPACT TRANSITION:
C                              LOWER LEVEL STATISTICAL WEIGHT
C                             1ST DIMENSION - TRANSITION NUMBER
C  INPUT : (I*4)  W2A()    = ELECTRON IMPACT TRANSITION:
C                              UPPER LEVEL STATISTICAL WEIGHT
C                             1ST DIMENSION - TRANSITION NUMBER
C
C  INPUT : (I*4)  IEC1A() = TRANSITION ENERGY INTERPOLATION VARIABLE
C                             (1=>Z1 ; 2=>1/Z1)
C                             1ST DIMENSION - TRANSITION NUMBER
C  INPUT : (I*4)  IAC1A() = TRANSITION PROB. INTERPOLATION VARIABLE
C                             (1=>Z1 ; 2=> 1/Z1)
C                             1ST DIMENSION - TRANSITION NUMBER
C  INPUT : (I*4)  IAC2A() = TRANSITION TYPE
C                             (1=>DIPOLE, 2=>NON-DIPOLE, 3=>SPIN CHANGE,
C                              4=>OTHER)
C                             1ST DIMENSION - TRANSITION NUMBER
C  INPUT : (I*4)  FAC2A() = TRANSITION PROB. Z1 SCALING POWER
C                             1ST DIMENSION - TRANSITION NUMBER
C  INPUT : (I*4)  IGC1A() = UPSILON INTERPOLATION VARIABLE
C                             (1=>Z1 ; 2=> 1/Z1)
C                             1ST DIMENSION - TRANSITION NUMBER
C  INPUT : (I*4)  FGC2A() = UPSILON Z1 SCALING POWER
C                             1ST DIMENSION - TRANSITION NUMBER
C  INPUT : (C*18) CTSTRA()= TRANSITION DESCRIPTOR
C                             1ST DIMENSION - TRANSITION NUMBER
C  INPUT : (R*8)  WDEA()  = TRANSITION ENERGY (CM-1)
C                             1ST DIMENSION - TRANSITION NUMBER
C
C  INPUT : (R*8)  AVALA() = ELECTRON IMPACT TRANSITION:
C                              A-VALUES (SEC-1)
C                             1ST DIMENSION - TRANSITION NUMBER
C                             2ND DIMENSION - SEQUENCE MEMBER INDEX
C  INPUT : (R*8)  SCOMA(,,)=ELECTRON IMPACT TRANSITION:
C                              GAMMA VALUES
C                             1ST DIMENSION - TEMPERATURE 'SCEF()'
C                             2ND DIMENSION - TRANSITION NUMBER
C                             3RD DIMENSION - SEQUENCE MEMBER INDEX
C
C  INPUT : (I*8)  MAXT    = NUMBER OF OUTPUT TEMPERATURES
C  INPUT : (R*8)  TEA()   = OUTPUT TEMPERATURES (K)
C
C  OUTPUT: (R*8)  WDE()   = ENERGY OF TRANSITIONS (CM-1)
C                             1ST DIMENSION - TRANSITION NUMBER
C  OUTPUT: (R*8)  AVAL()  = A-VALUE OF TRANSITIONS (SEC-1)
C                             1ST DIMENSION - TRANSITION NUMBER
C  OUTPUT: (R*8)  SCOM(,) = SELECTED TRANSITION GAMMA VALUES:
C                              GAMMA VALUES
C                             1ST DIMENSION - OUTPUT TEMPERATURE
C                             2ND DIMENSION - TRANSITION NUMBER
C
C
C          (I*4)  I       = GENERAL USE.
```

```
C          (I*4)  J       = GENERAL USE.
C          (I*4)  J1      = INPUT DATA FILE - SELECTED TRANSITION:
C                           LOWER ENERGY LEVEL INDEX (CASE ' ' & 'P')
C          (I*4)  J2      = INPUT DATA FILE - SELECTED TRANSITION:
C                           UPPER ENERGY LEVEL INDEX (CASE ' ' & 'P')
C                           CAPTURING    LEVEL INDEX (CASE 'H' & 'R')
C
C          (R*8)  AVALM   = INPUT DATA FILE - SELECTED TRANSITION:
C                          MANTISSA OF:   ('IAPOW' => EXPONENT)
C                           A-VALUE (SEC-1)          (CASE ' ')
C                           NEUTRAL BEAM ENERGY      (CASE 'H')
C                           NOT USED                 (CASE 'P' & 'R')
C
C    (R*8)  EIJMOD  = MODULUS OF EIJ
C
C          (C*80) CLINE   = CURRENT ENERGY LEVEL INDEX PARAMETER LINE
C
C          (L*4)  LDATA   = IDENTIFIES  WHETHER  THE END OF AN  INPUT
C                           SECTION IN THE DATA SET HAS BEEN LOCATED.
C                           (.TRUE. => END OF SECTION REACHED)
C
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          B2SORT     ADAS      PERFORMS BUBBLE SORT OF 2 REAL ARRAYS
C
C AUTHOR:  H. P. SUMMERS, JET
C          K1/1/57
C          JET EXT. 4941
C
C DATE:    17/08/94
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 06-03-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - PUT UNDER S.C.C.S. CONTROL
C
C VERSION: 1.2                          DATE: 06-03-96
C MODIFIED: TIM HAMMOND
C               - REPLACED CALLS TO NSORT ROUTINE WITH CALLS TO B2SORT.
C                 NSORT IS USED TO SORT A REAL ARRAY AND ASSOCIATED
C                 INTEGER ARRAY WHEREAS WHAT WAS BEING PASSED TO IT WAS
C                 A REAL ARRAY AND ANOTHER, ASSOCIATED REAL ARRAY. B2SORT
C                 TAKES 2 REAL ARRAYS AS INPUT AND PERFORMS A BUBBLE SORT
C                 ON THEM.
C
C VERSION: 1.3                          DATE: 01-04-96
C MODIFIED: TIM HAMMOND
C               - REMOVED SUPERFLUOUS VARIABLES
C
C VERSION: 1.4                          DATE: 01-04-96
C MODIFIED: TIM HAMMOND
C               - CORRECTED MINOR SYNTAX ERROR
C
C VERSION: 1.5                          DATE: 23-05-96
C MODIFIED: WILLIAM OSBORN + HUGH SUMMERS
C               - REPLACED EIJ BY MOD(EIJ) IN DETERMINING LINE STRENGTHS
C   AND UPSILON FIT
C
```

```
C VERSION: 1.6                          DATE: 11-04-07
C MODIFIED: ALLAN WHITEFORD
C               - RENAMED SOLVE SUBROUTINE TO B3SOLV.
C
C
C VERSION: 1.6                          DATE: 23-04-07
C MODIFIED: ALLAN WHITEFORD
C               - RENAMED FORM SUBROUTINE TO B3FORM.
C               - RENAMED FORM2 SUBROUTINE TO B3FORM2.
C
C-----------------------------------------------------------------------
      CHARACTER*18        CTSTRA(NDTRN)
      INTEGER             I1A(NDTRN),  I2A(NDTRN),  IAC1A(NDTRN)
      INTEGER             IAC2A(NDTRN),             IEC1A(NDTRN)
      INTEGER             IGC1A(NDTRN),             ISTRN,       ITRAN
      INTEGER             IZ0,         IZDIMD,      IZMAX,       IZS
      INTEGER             MAXT,        N1A(NDTRN),  N2A(NDTRN),  NDTEM
      INTEGER             NDTRN,       NVA(NDTRN)
      LOGICAL             LIBPT
      REAL*8              AVAL(NDTRN), AVALA(NDTRN,IZDIMD)
      REAL*8              FAC2A(NDTRN),             FGC2A(NDTRN)
      REAL*8              SCEFA(NDTEM,NDTRN),       SCOM(NDTEM,NDTRN)
      REAL*8              SCOMA(NDTEM,NDTRN,IZDIMD)
      REAL*8              TEA(NDTEM),  W1A(NDTRN),  W2A(NDTRN)
      REAL*8              WDE(NDTRN),  WDEA(NDTRN,IZDIMD)
      REAL*8              Z1A(IZDIMD)
```

3.20 b4datd: Subroutine b4datd from library adas2xx

```
       SUBROUTINE B4DATD ( XRMEMB    , NPMNCL , IMAXX  ,
     &                     NREPX     , MAXTM  , TEM    ,
     &                     NDBFILM   , NBFIL  , NCUTMC ,
     &                     AUGM      , DRM    , DRMSF  ,
     &                     PWSAT     , DSNXRT , OPEN17 ,
     &                     dsnin     , adas_c , adas_u
     &                   )
C-----------------------------------------------------------------------
C
C  ************** FORTRAN77 SUBROUTINE: B4DATD  ************************
C
C  VERSION:  1.1
C
C  PURPOSE:  PROCESS  DIELECTRONIC DATA FILES TO PREPARE
C            DIELECTRONIC AND AUGER DATA FOR ADAS204
C
C            THE DR FILE LAYOUT IS SPECIFIED BY THE ADF09 FORMAT
C
C  DATA:     THE SOURCE DATA IS ACCESSED THROUGH A CROSS-REFERENCE FILE
C                 /../adas/adf18/a09_p204/<ion>n.dat
C            WHERE <ION> DENOTES THE RECOMBINED ION (EG. C4)
C
C            THE PARENT CROSS-REFERENCING IS BASED ON THE ADAS204
C            DRIVING INPUT DATA FILE SPECIFIED BY THE ADF25 FORMAT
C                 /../adas/adf25/bns<yr>#<seq>/bns<yr>#<seq>_<code>.dat
C            WHERE <yr>   IS A TWO DIGIT YEAR NUMBER
C                  <seq>  IS THE ISO=ELECTRONIC SEQUENCE SYMBOL
C                  <code> IS AN ION CODE (eg. c4) OR ELEMENT CODE
C                         (EG. c ) IF A NUMBER OF IONS OF THE
C                         ISO-ELECTRONIC SEQUENCE ARE STACKED
C                         SEQUENTIALLY.
C
C            THE FILE NAMES ARE ANALYSED BY ADAS204 AND WARNINGS ISSUED
C            IF APPROPRIATE.  THESE WARNINGS ARE NOT NECESSARILY FATAL.
C            FOR EXAMPLE, THE ADF18 FILE CONTAINS THE NAME OF ITS
C            EXPECTED DRIVING ADF25 FILE.  THESE DIFFER IF THE ADF25
C            FILE IS DRIVING A COMPLETE ISO-ELECTRONIC SEQUENCE CALC.
C            RATHER THAN JUST A SINGLE ION CASE.
C
C
C  INPUT:  (C*8)  XRMEMB  = CROSS-REFERENCE PARTITIONED DATA SET MEMBER
C          (I*4)  IMAXX   = NUMBER OF REPRESENTATIVE LEVELS IN THE
C                               EXTENDED SET REQUIRED FOR THE MAIN CODE
C          (I*4)  NREPX() = REPRESENTATIVE N-SHELLS FOR THE MAIN CODE
C          (I*4)  NPMNCL  = NUMBER OF PARENTS INCLUDED IN THE MAIN CODE
C                               ( GIVEN BY THE <INMEMB> FILE )
C          (I*4)  MAXTM   = NUMBER OF TEMPERATURES USED IN MAIN CODE
C          (R*8)  TEM()   = TEMPERATURES (K) USED IN THE MAIN CODE
C          (I*4)  NDBFILM = PARAMETER = MAXIMUM NUMBER OF DR FILES
C                                       MUST BE GREATER THAN NDBFIL
C          (C*120)DSNXRT  = FIRST PART OF CROSS REFERENCE FILE NAME
C      (L)    OPEN17  = .FALSE. -OUTPUT TO UNIT=17 SWITCHED OFF.
C
C  OUTPUT: (I*4)  NCUTMC(,) = N-SHELL CUT FOR AUGER RATES (AUGER CHANNEL
C                             OPENS AT NCUTMC+1)
C                                1ST. INDEX = INITIAL PARENT
C                                2ND. INDEX = FINAL PARENT
C          (R*8)  AUGM(,,,) = AUGER RATES (SEC-1)
C                             1ST INDEX =  REPRESENTATIVE LEVEL
```

```
C                                2ND INDEX =  INITIAL PARENT
C                                3RD INDEX =  INITIAL SPIN SYSTEM
C                                4TH INDEX =  FINAL PARENT
C          (R*8)  DRM(,,,,) = DIELECTRONIC RATE COEFFTS. (CM3 SEC-1)
C                                1ST INDEX =  REPRESENTATIVE LEVEL
C                                2ND INDEX =  TEMPERATURE
C                                3RD INDEX =  INITIAL PARENT
C                                4TH INDEX =  INITIAL SPIN SYSTEM
C                                5TH INDEX =  FINAL PARENT
C          (I*4)  NBFIL     = NUMBER OF DR FILES
C
C  PROGRAM:(I*4)  NDREP    = PARAMETER = MAXIMUM NUMBER OF
C                                        REPRESENTATIVE LEVELS
C          (I*4)  NDPRT    = PARAMETER = MAXIMUM NUMBER OF PARENTS
C          (I*4)  NDSYS    = PARAMETER = MAXIMUM NUMBER OF SPIN SYSTEMS
C          (I*4)  NDT      = PARAMETER = MAXIMUM NUMBER OF TEMPERATURES
C          (I*4)  NDBFIL   = PARAMETER = MAXIMUM NUMBER OF DR FILES
C          (I*4)  NDPAIR   = PARAMETER = MAXIMUM NUMBER OF AUGER RATE
C                                        PARENT PAIRS
C          (I*4)  NDREP    = PARAMETER = MAXIMUM NUMBER OF MAIN CODE
C                                        REPRESENTATIVE LEVELS
C          (I*4)  NDBREP   = PARAMETER = MAXIMUM NUMBER OF DR
C                                        REPRESENTATIVE LEVELS
C
C          (C*1)  CHARS1   = ONE CHARACTER
C          (C*4)  CHARS4   = FOUR CHARACTERS
C          (C*120)DSNBD()  = DR DIELECTRONIC DATA FILE MEMBER NAMES
C          (C*30) BPDS     = DR PARENT STATE DESCRIPTOR
C          (C*30) BPDSC()  = DR PARENT STATE DESCRIPTOR ARRAY
C          (C*120)DSNMC    = MAINCL CODE INPUT FILE MEMBER NAME
C          (C*120)DSNMCO   = MAINCL CODE OUTPUT FILE MEMBER NAME
C          (C*120)DSN      = CHARACTER FILE NAME WORKSPACE
C          (C*120)DSHORT   = CURRENT FILE NAME WITH SYMBOLIC NAMES
C          (C*8)  MEMBER   = FILE MEMBER NAME WORKSPACE
C          (C*80) STRING   = LINE OUT STRING
C          (C*133)LSTRNG   = LINE IN STRING
C          (C*89) LSTRGO   = LONG LINE OUT STRING
C          (L*4)  OPEN12   = 'TRUE' IF OPEN
C          (L*4)  OPEN13   = 'TRUE' IF OPEN
C          (L*4)  OPEN14   = 'TRUE' IF OPEN
C          (L*4)  LEXIST   = 'TRUE' IF FILE EXISTS
C          (L*4)  LSJ      = 'TRUE' IF FILE EXISTS
C          (L*4)  LSETX    = 'TRUE' IF SPLINE UNINITIATED
C
C          (I*4)  I          = RUNNING INDEX
C          (I*4)  IBDPA()    = PARENT INDEX IN THE COMPLETE DR LIST
C          (I*4)  IBFIL      = RUNNING INDEX FOR DR FILES
C          (I*4)  IBREP      = RUNNING REPRESENTATIVE SHELL INDEX
C          (I*4)  IBMAX()    = NUMBER OF DR REPRESENTATIVE LEVELS
C                                   1ST. INDEX = DR FILE INDEX
C          (I*4)  IBPR       = CURRENT PARENT READ FROM DR FILE
C          (I*4)  IBPRIA(,)  = INITIAL PARENT INDEX FROM LIST FOR A FILE
C                                   1ST. INDEX = LEVEL INDEX
C                                   2ND. INDEX = DR FILE INDEX
C          (I*4)  IBPRFA(,)  = FINAL   PARENT INDEX FROM LIST FOR A FILE
C                                   1ST. INDEX = LEVEL INDEX
C                                   2ND. INDEX = DR FILE INDEX
C          (I*4)  IBREP      = RUNNING INDEX FOR REPRESENTATIVE LEVELS
C          (I*4)  IC         = COUNTER OF N-SHELLS BELOW AUGER CUT
C          (I*4)  IF         = RUNNING INDEX ON TOTAL PARENT LIST
C          (I*4)  II         = RUNNING INDEX ON TOTAL PARENT LIST
```

```
C          (I*4)  IMNPA()   = PARENT INDEX CORRESPONDING TO MAIN CODE
C          (I*4)  IND       = CHARACTER INDEX POSITION MARKER ON STRING
C          (I*4)  IOPT      =  SPLINE END CONDITION OPTION (SET =-1)
C          (I*4)  IP        = RUNNING INDEX ON TOTAL PARENT COUNT FROM
C                              DR FILES
C          (I*4)  IPI       = INITIAL PARENT OF SUPPL. AUGERING STATE
C          (I*4)  IPF       = FINAL PARENT AFTER SUPPL. AUGER
C          (I*4)  ISYSI     = INITIAL SPIN INDX. OF SUPPL.AUGERING STATE
C          (I*4)  IS        = RUNNING INDEX
C          (I*4)  ISREP     = SUPPLEMENTARY REPRESENTATIVE LEVEL INDEX
C          (I*4)  ISUPPLE   = NUMBER OF SUPPLE. AUGER RATES
C          (I*4)  IPAIRS    = RUNNING INDEX FOR AUGER RATE PARENT PAIRS
C          (I*4)  IPARM1    = DR FILE PARAMETER - PRTI
C          (I*4)  IPARM2    = DR FILE PARAMETER - TRMPRT
C          (I*4)  IPARM3    = DR FILE PARAMETER - SPNPRT
C          (I*4)  IPARM4    = DR FILE PARAMETER - PRTF
C          (I*4)  IPARM5    = DR FILE PARAMETER - TRMPRT
C          (I*4)  IPARM6    = DR FILE PARAMETER - SPNPRT
C          (I*4)  IPARM7    = DR FILE PARAMETER - NSYS
C          (I*4)  IPARM8    = DR FILE PARAMETER - SYS
C          (I*4)  IPARM9    = DR FILE PARAMETER - SPNSYS
C          (I*4)  IPRT      = RUNNING INDEX FOR PARENTS
C          (I*4)  IPT       = RUNNING INDEX ON TOTAL PARENT COUNT FROM
C                              DR FILES
C          (I*4)  IR        = UNSPECIFIED LINE COUNTER
C          (I*4)  IREAD     = FLAG FOR READ OPTION
C          (I*4)  IREFI()   = INITIAL PARENT FOR AUGER RATE IN FULL LIST
C          (I*4)  IREFF()   = FINAL PARENT FOR AUGER RATE IN FULL LIST
C          (I*4)  IREP      = MAIN CODE REPRESENTATIVE LEVEL COUNTER
C          (I*4)  IRFF      = POINTER TO FINAL PARENT IN FULL LIST
C          (I*4)  IRFI      = POINTER TO INITIAL PARENT IN FULL LIST
C          (I*4)  IS        = SPIN SYSTEM INDEX
C          (I*4)  ISET(,,)   = FLAG FOR INPUT OF SUPP. AUGER DATA
C                                       ISET = 0  NO SUPP. DATA
C                                       ISET = 1  SUPP. DATA
C                                       1ST INDEX - IPRT
C                                       2ND INDEX - ISYS
C                                       3RD INDEX - JPRT
C          (I*4)  ISPF      = FINAL PARENT SPIN FOR AUGER RATE
C          (I*4)  ISPFA(,)  = FINAL PARENT SPIN FOR AUGER RATE
C                                  1ST. INDEX = AUGER PARENT PAIR
C                                  2ND. INDEX = DR FILE INDEX
C          (I*4)  ISPI      = INITIAL PARENT SPIN FOR AUGER RATE
C          (I*4)  ISPIA(,)  = FINAL PARENT SPIN FOR AUGER RATE
C                                  1ST. INDEX = AUGER PARENT PAIR
C                                  2ND. INDEX = DR FILE INDEX
C          (I*4)  IST1    = PARAMETER = MAIN OUTPUT STREAM
C          (I*4)  ISYS      = RUNNING INDEX FOR SPIN SYSTEMS
C          (I*4)  IT        = RUNNING INDEX FOR TEMPERATURES
C          (I*4)  JPRT      = RUNNING INDEX FOR PARENTS
C          (I*4)  LEN1      = FIRST NON-BLANK CHARACTER IN MEMBER NAME
C          (I*4)  LEN2      = LAST NON-BLANK CHARACTER IN MEMBER NAME
C          (I*4)  MP()      = INITIAL PARENT INDEX FOR AUGER RATE
C          (I*4)  MPA()     = FINAL PARENT INDEX FOR AUGER RATE
C          (I*4)  NBCUT(,)  = N-SHELL CUT FOR AUGER RATES (AUGER CHANNEL
C                              OPENS AT NBCUT+1)
C                                  1ST. INDEX = AUGER PARENT PAIR
C                                  2ND. INDEX = DR FILE INDEX
C          (I*4)  NBFIL     = NUMBER OF DR FILES TO BE INCLUDED
C          (I*4)  NBREP(,)  = DR REPRESENTATIVE LEVEL N -VALUE
C                                  1ST. INDEX = LEVEL INDEX
```

```
C                                2ND. INDEX = DR FILE INDEX
C          (I*4)  NBT       = NUMBER OF DR TEMPERATURES
C          (I*4)  NCUTS     = FIRST OPENING NSHELL FOR SUPPL. AUGER
C          (I*4)  NDAUG     = PARAMETER = MAXIMUM N-SHELL OF SPECIFIC
C                                AUGER DATA
C          (I*4)  NPAIRS    = NUMBER OF AUGER RATE PARENT PAIRS
C          (I*4)  NPRNT     =
C          (I*4)  NPRNTF()  = NUMBER OF FINAL DR PARENTS FOR FILE
C          (I*4)  NPRNTI()  = NUMBER OF INITIAL DR PARENTS FOR FILE
C          (I*4)  NPTOT     = TOTAL NUMBER OF PARENTS ACCUMULATED FROM
C          (I*4)  NREP      = VALUE OF REPRESENTATIVE N-SHELL NREPX(IREP)
C                             DR FILES
C          (I*4)  NSREP()   = SUPPLEMENTARY AUGER REPRESENT. N-SHELLS
C          (I*4)  NTOP      = MARKS DRM ARRAY ZERO FOR N>NTOP
C
C          (R*8)  AA()      = SET OF AUGER RATES ON A LINE
C          (R*8)  AAS       = SUPPL. AUGER COEFFT. AT NCUTS (SEC-1)
C          (R*8)  AUGTMP(N)  = TEMPORARY STORE OF SUPP. AUGER RATES
C                                1ST INDEX - N-SHELL VALUE
C          (R*8)  DDRROUT() = SCALED DIELECTRONIC DATA FOR SPLINE IN N
C          (R*8)  DELTAE    = SATELLITE ENERGY LEVEL ( K)
C          (R*8)  DRRIN()   = SCALED DIELECTRONIC DATA FOR SPLINE IN N
C          (R*8)  DRMSF(,,,,)  SUMMED DR COEFFICIENT
C                                1ST INDEX - FILE
C                                2ND INDEX - TEMPERATURE
C                                3RD INDEX - INITIAL PARENT
C                                4TH INDEX - SPIN SYSTEM
C                                5TH INDEX - FINAL PARENT
C          (R*8)  DRMS()       TEMPORARY STORE OF SUMMED DR RATES
C                                1ST INDEX - TEMPERATURE
C          (R*8)  DRMF(,)      TEMPORARY STORE OF DR RATES FOR NBREP
C                                1ST INDEX - REPRESENTATIVE LEVEL
C                                2ND INDEX - TEMPERATURE
C          (R*8)  DTMP()    = TEMPORARY STORE OF DIEL. COEFFICIENTS
C          (R*8)  DRROUT()  = SCALED DIELECTRONIC DATA FOR SPLINE IN N
C          (R*8)  DY()      = WORK VECTOR FOR SPLINE
C          (R*8)  SLOPE     = N POWER FOR SUPPL. AUGER RATE ABOVE NCUTS
C          (R*8)  SYSFAC(,) = SPIN SYSTEM RESOLUTION OF AUGER RATES
C                               1ST. INDEX = AUGER RATE INDEX ON LINE
C                               2ND. INDEX = SPIN SYSTEM
C          (R*8)  TEB()     = DR TEMPERATURES (K)
C          (R*8)  XIN()     = WORK VECTOR FOR SPLINES
C          (R*8)  XOUT()    = WORK VECTOR FOR SPLINE
C          (R*8)  XNBREP()  = DR REPRES. LEVEL N -VALUE AS A REAL
C                               1ST. INDEX = LEVEL INDEX
C          (R*8)  XNREPX()  = REPRES. LEVEL N-VALUE FROM MAIN CODE AS A
C                                REAL
C          (R*8)  YIN()     = WORK VECTOR FOR SPLINES
C          (R*8)  YOUT()    = WORK VECTOR FOR SPLINE
C
C
C          (R*8)  XNREP    = REAL VARIABLE FORM OF NREP
C
C          (R*8)  XICENH   = IC ENHANCEMENT FACTOR FOR SPECIFIC
C                            N-SHELL
C
C
C  ROUTINES:
C           ROUTINE    SOURCE    BRIEF DESCRIPTION
C           ------------------------------------------------------------
C            B4FLNM      ADAS    EXPAND FILENAME SYMBOLIC PART IF PRESENT
```

```
C           B4SUMD      ADAS    SUMS DR COEFFICIENTS OVER ALL N-SHELLS
C           FINTB       HPS     CONVERTS X-VALUES FOR N SHELL SPINE
C           XXSLEN      ADAS    FINDS LENGTH OF NON-BLANK PART OF STRINGS
C           XXSPLN      ADAS    GENERAL CUBIC SPLINE
C
C
C  AUTHOR:  HUGH P. SUMMERS, JET
C           K1/1/57
C           JET EXT. 4941
C
C  DATE:    12/05/92
C
C  UPDATE:  04/06/92,  WILLIAM J. DICKSON , JET
C           ADJUSTED FORMAT STATEMENTS FROM ORIGINAL SPEC.
C           TO READ DR FILES WITH CHARACTERS SHIFTED ONE
C           SPACE TO THE LEFT.
C           DEFINED OUTPUT STREAM BY PARAMETER IST1
C
C  UPDATE:  07/92,  WILLIAM J. DICKSON , JET
C           DEFINE VALUE OF LSETX AT BEGINNING OF CODE
C
C  UPDATE:  27/08/92,  WILLIAM J. DICKSON , JET
C           (1)  ALLOW FOR SPECIFIC DATA FOR LOWEST N-SHELLS WHEN
C           INPUTING SUPPLEMENTARY AUGER TRANSITION PROBABILITIES
C           (2)  DEFINE VARIABLE ISET TO MARK SUPPLEMENTARY DATA INPUT
C
C  UPDATE:  06/09/92,  WILLIAM J. DICKSON , JET
C           XREF FILES NOW STORED UNDER JETXLE
C
C  UPDATE:  14/12/92,  WILLIAM J. DICKSON , JET
C           SET UP ROUTINE TO SUM DR COEFFICIENTS OVER
C           REPRESENTATIVE SET
C  UPDATE:  13/11/93,  WILLIAM J. DICKSON , JET
C           (1)  ALLOW FOR IC ENHANCEMENT FACTOR TO BE READ IN AS PART
C                FILE AND SUBSEQUENT ADJUSTMENT OF DR RATE COEFFICIENT
C                  CHECK CODING AROUND FORMAT STATEMENT 1036.
C                  (NOTE THAT 1037 WAS ADDED AT THIS STAGE)
C
C  UPDATE:  29/05/96  HP SUMMERS - COMPLETED UNIX FILE NAME PROCUREMENT
C                                  WITH ENVIRONMENT VARIABLE SYMBOL
C                                  SUBSTITUTION USING B4FLNM
C  UPDATE:  22/01/97  HP SUMMERS - CHANGED NAME TO B4DATD FROM BDMNCL1
C                                  AND SUBROUTINE BDDRSM2 TO B4SUMD
C  UPDATE:  11/02/97  HP SUMMERS - IMPROVED INTERPOLATION OF SUPPLE.
C                                  AUGER DATA FROM X-REF FILE.
C  UPDATE:  17/02/97  HP SUMMERS - IMPROVED INTERPOLATION OF DR. DATA
C                                  WITH N, TO ENSURE ABSOLUTE ZEROS
C                                  ABOVE CUT-OFF N-SHELL
C-----------------------------------------------------------------------
C
C UNIX-IDL CONVERSION:
C
C VERSION: 1.1 DATE: 05-03-98
C MODIFIED: H. SUMMERS
C     - MODIFIED VESION OF BDMNCL1.FOR v 1.1
C
C VERSION: 1.2  DATE: 26-11-98
C MODIFIED: Martin O'Mullane
C                  - redefine DSNXRT as the full DR supplement file
C                    name. It is now given in the adf25 dataset and
C                    passed through to here.
```

```
C
C VERSION: 1.3  DATE: 22-09-2000
C MODIFIED: Martin O'Mullane
C                   - Initialize ibmax to 0 to avoid troubles in the
C                     H-like case where we have no DR.
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      CHARACTER*80        ADAS_C,      ADAS_U
      CHARACTER*120       DSNIN,       DSNXRT
      CHARACTER*8         XRMEMB
      INTEGER             IMAXX,       MAXTM,       NBFIL
      INTEGER             NCUTMC(NDPRT,NDPRT),      NDBFILM,     NPMNCL
      INTEGER             NREPX(NDREP)
      LOGICAL             OPEN17
      REAL*8              AUGM(NDREP,NDPRT,NDSYS,NDPRT)
      REAL*8              DRM(NDREP,NDT,NDPRT,NDSYS,NDPRT)
      REAL*8              DRMSF(NDBFILM,NDT,NDPRT,NDSYS,NDPRT)
      REAL*8              PWSAT(NDBFILM,NDT,NDPRT,NDSYS,NDPRT)
      REAL*8              TEM(NDT)
```

3.21 b4flnm: Subroutine b4flnm from library adas2xx

```
       SUBROUTINE B4FLNM ( ADAS_C, ADAS_U, DSNIN, DSNFUL, LEXIST)
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: B4FLNM *********************
C
C  PURPOSE: TO PREPARE A UNIX DATASET NAME FROM A STRING WHICH MAY
C           INCLUDE AN ADAS ENVIRONMENT VARIABLE AND COMMENTS.
C
C           THE ADAS ENVIRONMENT VARIABLE MUST BE FIRST AND IN DOUBLE
C           QUOTES.  THE COMMENTS MUST EITHER FOLLOW A COLON.
C
C  CALLING PROGRAM: GENERAL USE
C
C  SUBROUTINE:
C
C  INPUT : (C120) DSNIN    = INPUT STRING FOR INTERROGATION
C  INPUT : (C*80) ADAS_C   = CENTRAL ADAS LOCATION (FROM IDL)
C  INPUT : (C*80) ADAS_U   = USER ADAS LOCATION (FROM IDL)
C
C  OUTPUT: (C120) DSNFUL   = THE FULL EXPANDED FILE NAME WITHOUT
C                            EXTRANEOUS MATERIAL
C  OUTPUT: (L*4)  LEXIST   = .TRUE. => NAME FORMED AND FILE EXISTS
C                            .FALSE.=> FAILED TO FORM NAME OR FIND FILE
C
C          (C*120) DSN1     = WORK STRING
C          (C*120) DSNTEMP  = WORK STRING
C          (C*120) BLANK    = BLANK STRING
C          (I*4)   LEN1     = STRING INDEX
C          (I*4)   LEN2     = STRING INDEX
C          (I*4)   LEN3     = STRING INDEX
C          (I*4)   LEN4     = STRING INDEX
C
C ROUTINES:
C          ROUTINE    SOURCE   BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          XXSLEN     ADAS     FIND BEGINNING AND END OF A STRING
C          I4UNIT     ADAS     FETCH UNIT NUMBER FOR OUTPUT OF MESSAGES
C
C AUTHOR:  H. P. SUMMERS, UNIVERSITY OF STRATHCLYDE
C          JA8.08
C          TEL. 0141-553-4196
C
C DATE  :  22/08/96
C
C VERSION: 1.3                                          DATE: 3-12-98
C MODIFIED: M.O'MULLANE
C           - Pass in ADAS environment variables
C
C VERSION: 1.4                                          DATE: 20-7-07
C MODIFIED: Allan Whiteford
C           - Small modification to comments to allow for automatic
C             documentation preparation.
C
C-----------------------------------------------------------------------
      CHARACTER*80        ADAS_C,      ADAS_U
      CHARACTER*120       DSNFUL,      DSNIN
      LOGICAL             LEXIST
```

3.22 b4matv: Subroutine b4matv from library adas2xx

```
      SUBROUTINE B4MATV(A,N,B,M,DETERM)
       IMPLICIT REAL*8 (A-H,O-Z)
C-----------------------------------------------------------------------
C
C  ************** FORTRAN77 SUBROUTINE: B4MATV ************************
C
C  VERSION:  1.0
C
C  PURPOSE:
C     MATRIX INVERSION WITH ACCOMPANYING SOLUTION OF LINEAR EQUATIONS
C
C  THIS SUBROUTINE IS NOT YET PROPERLY DOCUMENTED
C
C-----------------------------------------------------------------------
C
C UNIX-IDL CONVERSION:
C
C VERSION: 1.1                          DATE: 05-03-98
C MODIFIED: RICHARD MARTIN
C               - RENAMED FROM V2MATIV.FOR V1.1
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
C
      INTEGER             M,           N
      REAL*8              A(NREPDM,NREPDM),         B(NREPDM),   DETERM
```

3.23 b4proj: Subroutine b4proj from library adas2xx

```
       SUBROUTINE B4PROJ ( W1     , JTE    , JDENS  ,
     &                     NMIN   , NMAX   , NREP   , IMAX   ,
     &                     NRESU  , ARED   , RHS    , CIONPT ,
     &                     TRECPT , DRECPT , RRECPT , XRECPT ,
     &                     NPRT   , NMAXI  , NREPI  ,
     &                     IMAXI  , AREDI  , RHSI   ,
     &                     CIONRI , CIONRA , RHSIRC ,
     &                     IEDMAT , IECION , IETREC ,
     &                     IEDREC , IERREC , IEXREC ,
     &                     IERSYS , SSYSWT , IPRTCAL,
     &                     DVEC   , ACNST  , A1CNST ,
     &                     OPEN18 , OPEN19 , OPEN20 ,
     &                     PRB
     &                   )

C-----------------------------------------------------------------------
C
C  ************** FORTRAN77 SUBROUTINE: B4PROJ  ************************
C
C  VERSION:  1.1
C
C  PURPOSE: SUBROUTINE TO ESTABLISH THE PROJECTED INFLUENCE OF HIGH
C           N-SHELLS IN THE BUNDLE-N COLLISIONAL DIELECTRONIC MODEL
C           ON LOW N-SHELLS
C
C  BOTH THE RECOMBINATION AND IONISATION PATHWAYS THROUGH THE HIGH
C  LEVELS ARE TAKEN INTO ACCOUNT AS WELL AS THE INDIRECT COUPLINGS OF
C  LOW RESOLVED LEVELS VIA THE HIGH BUNDLE-N LEVELS.
C
C  THE SUBROUTINE IS USED AS AN ARBITRARY CALL FROM WITHIN THE
C  CONVENTIONAL BNDLEN ROUTINE FOLLOWING ESTABLISHMENT OF THE
C  CONDENSED COLLISIONAL-DIELECTRONIC MATRIX AND RIGHT-HAND SIDE
C
C  THE ROUTINE PROVIDES TABULAR OUTPUT AND FOR THE MOMENT PREPARES A
C  PASSING FILE FOR FURTHER PROCESSING IN THE A-D-A-S STRUCTURE
C
C
C  INPUT:
C      W1       = GROUND STATE RADIATION DILUTION FACTOR
C      JTE      = TEMPERATURE INDEX
C      JDENS    = DENSITY INDEX
C      NMIN     = LOWEST N-SHELL
C      NMAX     = HIGHEST N-SHELL
C      NREP(I)  = SET OF REPRESENTATIVE LEVELS
C      IMAX     = NUMBER OF REPRESENTATIVE LEVELS
C      NRESU    = UPPER LIMIT OF PROJECTED N-SHELLS
C
C      ARED(I,J)= CONDENSED COLLISONAL-DIELECTRONIC MATRIX (CN SOLUTION)
C                  (EXCLUDES AUTO-IONISATION RATES FOR LEVELS LE NRESU)
C      RHS(I)   = CONDENSED RIGHT-HAND-SIDE               (CN SOLUTION)
C                  (EXCLUDES AUTO-IONISATION RATES FOR LEVELS LE NRESU)
C      CIONPT(I)= COLLISIONAL IONISATION CONTRIBUTION TO ARED(I,I)
C      TRECPT(I)= THREE BODY   RECOMBINATION CONTRIBUTION TO RHS(I)
C      DRECPT(I)= DIELECTRONIC RECOMBINATION CONTRIBUTION TO RHS(I)
C      RRECPT(I)= RADIATIVE    RECOMBINATION CONTRIBUTION TO RHS(I)
C      XRECPT(I)= CHARGE EXCHANGE RECOMB.    CONTRIBUTION TO RHS(I)
C
C      NPRT     = NUMBER OF PARENT STATES
C      IMAXI
```

```
C      NMAXI
C      NREPI(I)   DATA FOR PROJECTION OF IONISATION VECTORS
C      AREDI(I,J)           SMALL (40X40) MATRIX , CN SOLUTION
C      RHSI(I)
C      RHSIRC(I)= RECOMBINATION CONTRIBUTION TO RHS
C      CIONRI   = DIRECT IONISATION DATA, PARENT RESOLVED
C      CIONRA   = AUTO-IONISATION DATA,   PARENT RESOLVED
C
C      SSYSWT   = SPIN SYSTEM WEIGHT
C      IPRTCAL  = INDEX OF PARENT FOR CALCULATION
C
C      DVEC(I)  = CONVERSION FACTOR FOR BN --> POPULATION
C      ACNST    = 1.03928D-13*Z*ATE*DSQRT(ATE)
C      A1CNST   = 6.60074D-24*DENS*(157890.0/TE)**1.5
C
C      PCION(I) = DIRECT IONISATION RATE FROM LOW LEVEL SET
C                          POPULATION REPRESENTATION
C      PRB      = RECOM/CASCADE/BREMS. POWER COEFFT.
C
C      OUTPUT - POPULATION REPRESENTATION (WRITTEN TO FILE CBNM.PASS)
C      ------------------------------------------------------------
C      PCRMAT(I,J) = PROJECTED INFLUENCE OF HIGH LEVELS ON LOW LEVEL SET
C      PCRL(I,J)   = DIRECT EXCIT/RADIATIVE COUPLING IN LOW LEVEL SET
C      PCIONRP(IPRT,I)  = PROJECTED IONISATION VECTOR (PARENT RESOLVED)
C      PCIONRI(IPRT,I)  = DIRECT IONISATION VECTOR FROM LOW LEVEL SET
C                                                      (PARENT RESOLVED)
C      PCQINRP(IPRT)    = INDIRECT PARENT CROSS COUPLING COEFFICIENT
C                                                      (PARENT RESOLVED)
C      PCRRHS(I)  = PROJECTED INFLUENCE OF HIGH LEVELS ON RHS
C      PTREC(I)   = DIRECT THREE BODY RECOMBINATION RATE
C      PDREC(I)   = DIRECT DIELECTRONIC RECOMBINATION RATE
C      PRREC(I)   = DIRECT RADIATIVE RECOMBINATION RATE
C      PXREC(I)   = DIRECT CX RECOMBINATION RATE
C      PRB        = RECOM/CASCADE/BREMS. POWER COEFFT.
C
C
C      OUPUT CONTROL CHARACTERS
C      ------------------------
C      IEDMAT   = 0 PCRL ADDED ONTO PCRMAT
C                 1 PCRL NOT ADDED ON
C      IECION   = 0 PCION  ADDED ONTO TO PCRMAT
C                   PCIONRI ADDED ONTO PCIONRP
C                 1 PCION NOT ADDED ON
C                   PCIONRI NOT ADDED ON
C      IETREC   = 0 PTREC ADDED ONTO PCRRHS
C                 1 PTREC NOT ADDED ON
C      IEDREC   = 0 PDREC ADDED ONTO PCRRHS
C                 1 PDREC NOT ADDED ON
C      IERREC   = 0 PRREC ADDED ONTO PCRRHS
C                 1 PRREC NOT ADDED ON
C      IEXREC   = 0 PXREC ADDED ONTO PCRRHS
C                 1 PXREC NOT ADDED ON
C      IERSYS   = 0 RECOMBINATION AND INDIRECT PARENT CROSS COUPLING
C                   RATES MULTIPLIED BY SPIN SYSTEM WEIGHT
C                 1 RECOMBINATION AND INDIRECT PARENT CROSS COUPLING
C                   RATES NOT MULTIPLIED BY SPIN SYSTEM WEIGHT
C
C
C  AUTHOR:  WILLIAM J. DICKSON, JET JOINT UNDERTAKING
C
C  DATE:  24TH AUGUST 1992
```

```
C
C  UPDATE:  30/01/97  HP SUMMERS - CHANGED NAME TO B4PROJ FROM V2CLDBN
C
C  UPDATE:  29/04/97  HP SUMMERS - ADJUSTMENTS DURING RE-VALIDATION
C
C  UPDATE:  09/07/97  HP SUMMERS - INTRODUCE IOUT18 AND IOUT19 FOR CBNM
C                                  AND CBNMPR PASSING FILES
C
C  UPDATE:  09/03/98  HP SUMMERS - RECOM/CASCADE/BREMS. POWER NOW
C                                  FETCHED AS INPUT PRB AND RELAYED TO
C                                  CBNM FILE.  CONVERTED TO EXPLICIT
C                                  TYPE DECLARATIONS.
C
C-----------------------------------------------------------------------
C
C  VERSION: 1.1 DATE: 05-03-98
C  MODIFIED: H.SUMMERS, L.HORTON, M.O'MULLANE
C - BASED ON v2cldbc.for v1.2.
C  VERSION: 1.2 DATE: 09-03-98
C  MODIFIED: H.SUMMERS, L.HORTON, M.O'MULLANE
C - RECOM/CASCADE/BREMS. POWER NOW FETCHED AS INPUT PRB AND
C   RELAYED TO CBNM FILE.  CONVERTED TO EXPLICIT TYPE DECLARATIONS.
C  VERSION: 1.3 DATE: 08-12-98
C HP SUMMERS & RICHARD MARTIN
C - REMOVED TWO OBSOLETE WRITE STATEMENTS.
C
C  VERSION: 1.4 DATE: 03-08-2000
C Martin O'Mullane
C - Changed IPRT\N to IPRT/N to avoid \N being interpreted
C                 as an escape character.
C
C  VERSION: 1.5 DATE: 11-09-2002
C Martin O'Mullane
C - Add open18 and open19 to give finer control over the
C                 output files; extra logic to use them added throughout.
C
C  VERSION: 1.6 DATE: 16-05-2007
C Allan Whiteford
C - Modified comments as part of subroutine documentation
C                 procedure.
C
C-----------------------------------------------------------------------
      INTEGER             IECION,      IEDMAT,      IEDREC,      IERREC
      INTEGER             IERSYS,      IETREC,      IEXREC,      IMAX
      INTEGER             IMAXI,       IPRTCAL,     JDENS,       JTE
      INTEGER             NMAX,        NMAXI,       NMIN,        NPRT
      INTEGER             NREP(NDIM+1),             NREPI(NDIM+1)
      INTEGER             NRESU
      LOGICAL             OPEN18,      OPEN19,      OPEN20
      REAL*8              A1CNST,      ACNST,       ARED(NDIM,NDIM)
      REAL*8              AREDI(NDIM,NDIM),         CIONPT(NDIM)
      REAL*8              CIONRA(NDMET,NDIM),       CIONRI(NDMET,NDIM)
      REAL*8              DRECPT(NDIM),             DVEC(NDIM),  PRB
      REAL*8              RHS(NDIM),   RHSI(NDIM),  RHSIRC(NDIM)
      REAL*8              RRECPT(NDIM),             SSYSWT
      REAL*8              TRECPT(NDIM),             W1
      REAL*8              XRECPT(NDIM)
```

3.24 b4spln: Subroutine b4spln from library adas2xx

```
      SUBROUTINE B4SPLN( ITA    , ITVAL   ,
     &                   BWNO   ,
     &                   TETA   , TEVA    ,
     &                   SZD    ,
     &                   SZDA   , ESZDA   ,
     &                   LTRNG
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: B4SPLN *********************
C
C  PURPOSE:
C          PERFORMS CUBIC SPLINE ON LOG(TEMPERATURE <EV> ) VERSUS
C          LOG(SCALED IONIZATION RATE COEFFICIENTS).
C          INPUT DATA FOR A GIVEN IONZING ION COMBINATION DATA-BLOCK.
C
C          USING  ONE-WAY SPLINES  IT  CALCULATES  THE  IONIZATION  RATE
C          COEFFICIENT  FOR  'ITVAL'  ELECTRON  TEMPERATURE VALUES  FROM
C          THE LIST OF ELECTRON TEMPERATURES READ IN FROM THE INPUT FILE
C
C          IF A  VALUE  CANNOT  BE  INTERPOLATED  USING  SPLINES  IT  IS
C          EXTRAPOLATED VIA 'XXSPLE'. (SEE NOTES BELOW).
C
C  CALLING PROGRAM: ADAS204/B4SSZD
C
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  ITA     = INPUT DATA FILE: NUMBER OF ELECTRON TEMPERA-
C                           TURES READ FOR THE DATA-BLOCK BEING ASSESSED
C  INPUT : (I*4)  ITVAL   = NUMBER OF ISPF ENTERED ELECTRON TEMPERATURE
C                           VALUES FOR  WHICH  IOINIZATION RATE COEFFTS
C                           ARE REQUIRED FOR TABULAR/GRAPHICAL OUTPUT.
C
C  INPUT : (R*8)  BWNO    = INPUT DATA FILE: IONIZATION POTENTIAL (cm-1)
C                           FOR THE DATA-BLOCK BEING ASSESSED.
C
C  INPUT : (R*8)  TETA()  = INPUT DATA FILE: ELECTRON TEMPERATURES (EV)
C                           FOR THE DATA-BLOCK BEING ASSESSED.
C                           DIMENSION: ELECTRON TEMPERATURE INDEX
C  INPUT : (R*8)  TEVA()  = USER ENTERED: ELECTRON TEMPERATURES (EV)
C                           DIMENSION: TEMPERATURE/DENSITY PAIR INDEX
C
C  INPUT : (R*8)  SZD()    =INPUT DATA FILE: FULL SET OF ZERO DENSITY
C                           IONIZATION RATE COEFFTS FOR THE DATA-BLOCK
C                           BEING ANALYSED.
C                           1ST DIMENSION: ELECTRON TEMPERATURE INDEX
C  OUTPUT: (R*8)  SZDA()  = SPLINE INTERPOLATED OR  EXTRAPOLATED  ZERO
C                           DENSITY IONIZATION RATE COEFFICIENTS FOR
C                           THE USER ENTERED ELECTRON TEMPERATURES.
C                           DIMENSION: ELECTRON TEMPERATURE INDEX
C  OUTPUT: (R*8)  ESZDA() = EXP((BWNO/109737.3)*(IH/KTE))*SZDA()
C
C  OUTPUT: (L*4)  LTRNG()=  .TRUE.  => OUTPUT 'SZDA()' VALUE WAS INTER-
C                                      POLATED  FOR  THE  USER  ENTERED
C                                      ELECTRON TEMPERATURE 'TEVA()'.
C                           .FALSE. => OUTPUT 'SZDA()' VALUE WAS EXTRA-
C                                      POLATED  FOR  THE  USER  ENTERED
C                                      ELECTRON TEMPERATURE 'TEVA()'.
```

```
C                          DIMENSION: ELECTRON TEMPERATURE INDEX
C
C          (I*4)  NIN     = PARAMETER = MAX. NO. OF INPUT  TEMPERATURE
C                                       VALUES. MUST BE >= 'ITA'
C          (I*4)  NOUT    = PARAMETER = MAX. NO. OF OUTPUT TEMPERATURE
C                                       PAIRS.  MUST BE >= 'ITVAL'
C          (I*4)  L1      = PARAMETER = 1
C
C          (R*8)  BCONST  = PARAMETER = 1/(SCALED BOLTZMANN'S CONSTANT)
C
C          (I*4)  IET     = ARRAY SUBSCRIPT USED INPUT  FILE  ELECTRON
C                           TEMPERATURES.
C          (I*4)  IT      = ARRAY  SUBSCRIPT  USED  FOR  USER  ENTERED
C                           TEMPERATURE VALUES.
C          (I*4)  IOPT    = DEFINES THE BOUNDARY DERIVATIVES FOR THE
C                           SPLINE ROUTINE 'XXSPLE', SEE 'XXSPLE'.
C                           (VALID VALUES = <0, 0, 1, 2, 3, 4)
C
C          (L*4)  LSETX   = .TRUE.  => SET UP SPLINE PARAMETERS RELATING
C                                      TO 'XIN' AXIS.
C                           .FALSE. => DO NOT SET UP SPLINE PARAMETERS
C                                      RELATING TO 'XIN' AXIS.
C                                      (I.E. THEY WERE SET IN A PREVIOUS
C                                            CALL )
C                           (VALUE SET TO .FALSE. BY 'XXSPLE')
C
C          (R*8)  R8FUN1  = FUNCTION - (SEE ROUTINES SECTION BELOW)
C          (R*8)  SCONST  = SCALING CONSTANT USED TO SCALE THE IONIZA-
C                           TION RATE COEFFT. WHEN SPLINNING.
C                           = IONIZATION POTENTIAL / BOLTZMANN CONST.
C
C          (R*8)  XIN()   = LOG( DATA FILE ELECTRON TEMPERATURES )
C          (R*8)  YIN()   = LOG( DATA FILE SCALED ION. RATE COEFFTS.)
C          (R*8)  XOUT()  = LOG( USER ENTERED ELECTRON TEMPS.)
C          (R*8)  YOUT()  = LOG( OUTPUT GENERATED SCALED ION. RATE COEF)
C          (R*8)  DF()    = SPLINE INTERPOLATED DERIVATIVES
C
C
C NOTE:
C
C          ONE-DIMENSIONAL SPLINE CARRIED OUT BY THIS SUBROUTINE:
C
C           LOG( EXP(<ion.pt.>/<k>.<Te>) . Szd )  vs. LOG( Te )
C
C           ion.pt. = ionization potential (units: cm-1)
C           k       = Boltzmann's constant (= 1/1.23977E-04)
C           Te      = electron temperature (units: eV)
C           Szd     = zero density ionization rate coefficient
C                                      (units: cm**3/sec)
C
C           Extrapolation criteria:
C
C           Low  Te: zero gradient  extrapolation (i.e. DY(1)  = 0.0)
C           High Te: zero curvature extrapolation (i.e. DDY(N) = 0.0)
C
C           (These criteria are met by calling XXSPLE with IOPT=4)
C
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
```

```
C          XXSPLE     ADAS      SPLINE SUBROUTINE (EXTENDED DIAGNOSTICS)
C          R8FUN1     ADAS      REAL*8 FUNCTION: ( X -> X )
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/37
C          JET EXT. 2620
C
C DATE:    07/06/91
C
C UPDATE:  17/02/97 HP SUMMERS - ADDED EXP(IP/KTE)*SZD AS AN OUTPUT
C                                PARAMETER.
C
C VERSION: 1.1 DATE: 05-03-98
C MODIFIED: H.SUMMERS, L.HORTON, M.OMULLANE, R.MARTIN
C     - BASED ON E2SPLN.FOR v1.2. PUT UNDER SCCS CONTROL.
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             ITA,          ITVAL
      LOGICAL             LTRNG(ITVAL)
      REAL*8              BWNO,         ESZDA(ITVAL)
      REAL*8              SZD(ITA),     SZDA(ITVAL), TETA(ITA)
      REAL*8              TEVA(ITVAL)
```

3.25 b4sszd: Subroutine b4sszd from library adas2xx

```
      SUBROUTINE B4SSZD( dsname , IBSEL  , IZ0IN  ,
     &                   ITVAL  , TVAL   ,
     &                   BWNO   , IZ     , IZ1  ,
     &                   METI   , METF   ,
     &                   SZDA   , ESZDA  , LTRNG  ,
     &                   TITLX  , IRCODE , OPEN17
     &                )
C-----------------------------------------------------------------------
C
C  ******************* FORTRAN77 SUBROUTINE: B4SSZD *******************
C
C  PURPOSE: TO EXTRACT AND  INTERPOLATE  ZERO-DENSITY  IONIZATION  RATE-
C           COEFFICIENTS FOR GIVEN ELEMENT NUCLEAR CHARGE AND DATA-BLOCK
C           FOR AN INPUT SET OF ELECTRON TEMPERATURES (eV).
C
C
C  CALLING PROGRAM: GENERAL USE
C
C  SUBROUTINE:
C
C  INPUT : (C*80) DSNAME  = ADF07 DATAFILE NAME UNDER UNIX INCLUDING PATH
C  INPUT : (I*4)  IBSEL   = INDEX OF DATA-BLOCK SELECTED FOR ANALYSIS
C  INPUT : (I*4)  IZ0IN   = NUCLEAR CHARGE OF REQUIRED ELEMENT
C
C  INPUT : (I*4)  ITVAL   = NUMBER OF ELECTRON TEMPERATURE VALUES
C  INPUT : (R*8)  TVAL()  = ELECTRON TEMPERATUIRES (UNITS: EV)
C                           DIMENSION: ELECTRON TEMPERATURE INDEX
C
C  OUTPUT: (R*8)  BWNO    = INPUT FILE - SELECTED DATA-BLOCK:
C                           EFFECTIVE IONIZATION POTENTIAL (cm-1).
C  OUTPUT: (I*4)  IZ      = INPUT FILE - SELECTED DATA BLOCK:
C                           IONIZING ION - INITIAL CHARGE
C  OUTPUT: (I*4)  IZ1     = INPUT FILE - SELECTED DATA BLOCK:
C                           IONIZING ION - FINAL   CHARGE
C
C  OUTPUT: (I*4)  METI    = INPUT FILE - SELECTED DATA-BLOCK:
C                           INITIAL STATE METSTABLE INDEX
C  OUTPUT: (I*4)  METF    = INPUT FILE - SELECTED DATA-BLOCK:
C                           FINAL   STATE METSTABLE INDEX
C
C  OUTPUT: (R*8)  SZDA()  = ZERO-DENSITY IONIZATION RATE-COEFFICIENTS
C                           DIMENSION: ELECTRON TEMPERATURE INDEX
C  OUTPUT: (R*8)  ESZDA() = EXP((BWNO/109737.3)*(IH/KTE))*SZDA()
C                           DIMENSION: ELECTRON TEMPERATURE INDEX
C  OUTPUT: (L*4)  LTRNG() =.TRUE.  => OUTPUT 'SZDA()'  VALUE WAS INTER-
C                                     POLATED  FOR  THE  USER  ENTERED
C                                     ELECTRON TEMPERATURE 'TVAL()'.
C                          .FALSE. => OUTPUT 'SZDA()'  VALUE WAS EXTRA-
C                                     POLATED  FOR  THE  USER  ENTERED
C                                     ELECTRON TEMPERATURE 'TVAL()'.
C                           DIMENSION: ELECTRON TEMPERATURE INDEX
C
C  OUTPUT: (C*120)TITLX   = INFORMATION STRING (DSN ETC.)
C  OUTPUT: (I*4)  IRCODE  = RETURN CODE FROM SUBROUTINE:
C                           0 => NORMAL COMPLETION - NO ERROR DETECTED
C                           2 => DISCREPANCY BETWEEN REQUESTED CHARGES
C                                AND THOSE IN INPUT FILE.
C                           3 => THE SELECTED DATA-BLOCK 'IBSEL' IS OUT
C                                OF RANGE OR DOES NOT EXIST.
```

```
C                          4 => INVALID VALUE FOR 'IZ0IN' ENTERED.
C                               ('IZ0MIN' <= 'IZ0IN' <= 'IZ0MAX')
C                          9 => ERROR ENCOUNTERED WHEN TRYING TO OPEN
C                               INPUT DATA-SET.
C
C          (I*4)  NSTORE  = PARAMETER= MAXIMUM NUMBER  OF  DATA-BLOCKS
C                                      WHICH CAN BE READ FROM THE INPUT
C                                      DATA-SET.
C          (I*4)  NTDIM   = PARAMETER= MAXIMUM NUMBER OF ELECTRON TEMP-
C                                      ERATURES THAT CAN BE READ  FROM
C                                      AN INPUT DATA-SET DATA-BLOCK.
C          (I*4)  IZ0MIN  = PARAMETER: MIN. ALLOWED VALUE FOR 'IZ0IN'
C          (I*4)  IZ0MAX  = PARAMETER: MAX. ALLOWED VALUE FOR 'IZ0IN'
C
C          (I*4)  IZ0LST  = LAST VALUE OF 'IZ0IN' FOR  WHICH  INPUT
C                           DATA WAS READ.
C          (I*4)  IUNIT   = UNIT TO WHICH INPUT DATA SET IS ALLOCATED
C          (I*4)  NBSEL   = TOTAL NUMBER OF DATA-BLOCKS READ FROM INPUT
C                           DATA SET.
C          (I*4)  IZ0     = INPUT FILE - EMITTING ION - NUCLEAR CHARGE
C
C          (L*4)  LOPEN   = .TRUE.  => INPUT DATA SET OPEN.
C                           .FALSE. => INPUT DATA SET CLOSED.
C
C          (C*2)  ESYM    = INPUT FILE - IONIZING ION - ELEMENT SYMBOL
C          (C*3)  EXTIN   = CURRENT ADAS SOURCE DATA FILE EXTENSION
C          (C*3)  EXTLST  = ADAS SOURCE DATA FILE EXT. USED LAST TIME
C                           DATA WAS READ.
C
C          (I*4)  ISELA() = INPUT DATA FILE: DATA-BLOCK ENTRY INDICES.
C                           DIMENSION: DATA-BLOCK INDEX
C          (I*4)  ITA()   = INPUT DATA SET-NUMBER OF ELECTRON TEMPERA-
C                           TURES.
C                           DIMENSION: DATA-BLOCK INDEX
C          (I*4)  IZOUT() = INPUT DATA FILE: IONIZING ION INITIAL CHARGE
C                           DIMENSION: DATA-BLOCK INDEX
C          (I*4)  IZ1OUT()= INPUT DATA FILE: IONIZING ION FINAL   CHARGE
C                           DIMENSION: DATA-BLOCK INDEX
C
C          (R*8)  BWNOUT()= INPUT DATA FILE: EFFECTIVE IONIZATION POT.
C                           (UNITS: cm-1).
C                           DIMENSION: DATA-BLOCK INDEX
C          (R*8)  TETA(,) = INPUT DATA SET -
C                           ELECTRON TEMPERATURES (UNITS: eV)
C                           1st DIMENSION: ELECTRON TEMPERATURE INDEX
C                           2nd DIMENSION: DATA-BLOCK INDEX
C          (R*8)  SZD(,)    =INPUT DATA SET -
C                            FULL SET OF IONIZATIONS RATE-COEFFICIENTS
C                            1st DIMENSION: ELECTRON TEMPERATURE INDEX
C                            3rd DIMENSION: DATA-BLOCK INDEX
C
C          (C*2)  CICODE()= INPUT DATA FILE - INITIAL STATE META. INDEX
C                           DIMENSION: DATA-BLOCK INDEX
C          (C*2)  CFCODE()= INPUT DATA FILE - FINAL   STATE META. INDEX
C                           DIMENSION: DATA-BLOCK INDEX
C          (C*6)  CIION() = INPUT DATA FILE - INITIAL ION
C                           DIMENSION: DATA-BLOCK INDEX
C          (C*6)  CFION() = INPUT DATA FILE - FINAL   ION
C                           DIMENSION: DATA-BLOCK INDEX
C
C
```

```
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          XXDATA_07  ADAS      FETCH INPUT DATA FROM SELECTED DATA SET
C          B4SPLN     ADAS      INTERPOLATE DATA WITH ONE-WAY SPLINES
C          E2TITL     ADAS      CREATE DESCRIPTIVE TITLE FOR OUTPUT
C
C
C Original version
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/37
C          JET EXT. 6023
C
C DATE:    07/06/91
C
C
C UPDATE:    17/02/97 - H P SUMMERS: RENAME SSZD AS B4SSZD. EXTRACT
C                                    Exp(I/KTE) * S AS WELL AS S
C
C UPDATE:    04-03-97 - R. MARTIN: ADDED OPEN17 FOR SWITCHING OUTPUT TO
C            'adas204.pass1' ON AND OFF.
C
C VERSION: 1.2                                          DATE: 03-12-98
C MODIFIED: Martin O'Mullane:
C                       Rewritten to account for adf07 filename
C                       being included in the adf25 namelist. It is
C                       now much simplified.
C
C VERSION: 1.3                                          DATE: 26-03-08
C MODIFIED: Allan Whiteford:
C                      Changed call from E2DATA to XXDATA_07.
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      CHARACTER*120       DSNAME,      TITLX
      INTEGER             IBSEL,       IRCODE,      ITVAL,       IZ
      INTEGER             IZ0IN,       IZ1,         METF,        METI
      LOGICAL             LTRNG(ITVAL),             OPEN17
      REAL*8              BWNO,        ESZDA(ITVAL)
      REAL*8              SZDA(ITVAL), TVAL(ITVAL)
```

3.26 b4sumd: Subroutine b4sumd from library adas2xx

```
       SUBROUTINE B4SUMD ( NDREP  , NDT     ,
     &                     MAXTM  , IREPMAX , IREP   , DRMF   , DRMS   ,
     &                     EIJN   , PWTEMP
     &                   )
C-----------------------------------------------------------------------
C
C *************** FORTRAN 77 SUBROUTINE: B4SUMD  *********************
C
C  VERSION: 2.0
C
C  PURPOSE: TO SUM BADNELL DIELECTRONIC RATE COEFFICIENT DATA OVER THE
C           REPRESENTATIVE SET TO GIVE ZERO DENSITY TOTAL AND
C           RADIATED POWER FROM SATELLITE LINES
C
C  CALLING PROGRAM: B4DATD
C
C
C  INPUT:
C  INPUT :  (I*4) NDREP    = MAXIMUM NUMBER OF REPRESENTATIVE LEVELS
C  INPUT :  (I*4) NDT      = MAXIMUM NUMBER OF TEMPERATURES
C  INPUT :  (I*4) DRMF(,)  = BADNELL DIELECTRONIC DATA (CM3 S-1)
C                            1ST DIM.: REPRESENTATIVE LEVEL INDEX
C                            2ND DIM.: TEMPERATURE INDEX
C  INPUT :  (I*4) NBT      = NO. OF TEMPERATURES
C  INPUT :  (I*4) IREPMAX  = NO OF REPRESENTATIVE LEVELS
C  INPUT :  (I*4) IREP()   = SET OF REPRESENTATIVE LEVELS
C  INPUT :  (R*8) EIJN()   = SATELLITE. ENERGY AS A FUNCTION OF
C                            REPRESENTATIVE LEVEL (K)
C
C  OUTPUT:  (R*8) DRMS()   = SUMMED DR RATE COEFFICIENTS (CM3 S-1)
C                            1ST DIM.: TEMPERATURE INDEX
C  OUTPUT:  (R*8) PWTEMP() = SAT. RADIATED POWER (UNITS ERG S-1 CM3)
C                            1ST DIM.: TEMPERATURE INDEX
C
C           (I*4) NREP     = GENERAL LEVEL INDEX
C           (I*4) IN       = GENERAL INDEX
C           (I*4) IT       = GENERAL INDEX
C           (R*8) V        = GENERAL VARIABLE FOR N-SHELL
C           (R*8) V1       = GENERAL VARIABLE FOR N-SHELL
C           (R*8) Y        = GENERAL VARIABLE FOR N-SHELL
C           (R*8) YP       = GENERAL VARIABLE FOR N-SHELL
C           (R*8) Y0       = GENERAL VARIABLE FOR N-SHELL
C           (R*8) Y1       = GENERAL VARIABLE FOR N-SHELL
C           (R*8) PW       = GENERAL VARIABLE FOR N-SHELL
C           (R*8) PW1      = GENERAL VARIABLE FOR N-SHELL
C
C
C  AUTHOR:   WILLIAM J. DICKSON, JET JOINT UNDERTAKING
C
C  DATE:     14TH DECEMBER 1992
C
C  UPDATE: 15/12/92  WJ DICKSON - REVISED ALGORITHM HAS BETTER
C                                 AGREEMENT WITH INTERNAL SUM
C                                 CALCULATED BY MAINCL
C
C  UPDATE: 31/01/97  HP SUMMERS - CHANGED NAME TO B4SUMD
C
C  VERSION: 1.1 DATE: 05-03-98
C  MODIFIED: RICHARD MARTIN
```

```
C - PUT UNDER SCCS CONTROL.
C
C-----------------------------------------------------------------------
      INTEGER             IREP(NDREP), IREPMAX,     MAXTM,       NDREP
      INTEGER             NDT
      REAL*8              DRMF(NDREP,NDT),          DRMS(NDT)
      REAL*8              EIJN(NDREP), PWTEMP(NDT)
```

3.27 b5wr12: Subroutine b5wr12 from library adas2xx

```
      SUBROUTINE B5WR12( IUNIT  , DATE   , IZ1   , IL    ,
     &                   NDMET  , NDTEM  , NDDEN ,
     &                   NMET   , IMETR  ,
     &                   IFOUT  , MAXT   , TINE  ,
     &                   IDOUT  , MAXD   , DINE  ,
     &                   CSTRGA , STCKM
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: B5WR12 *********************
C
C  PURPOSE: TO OUTPUT METASTABLE POPULATION PARAMETERS TO THE PASSING
C           FILE ON STREAM 'IUNIT'.
C
C  CALLING PROGRAM: ADAS205
C
C  SUBROUTINE:
C
C  INPUT :  (I*4)  IUNIT   = OUTPUT STREAM NUMBER
C  INPUT :  (C*8)  DATE    = CURRENT DATE AS 'DD/MM/YY'
C  INPUT :  (I*4)  IZ1     = RECOMBINING ION CHARGE
C                            (NOTE: IZ1 SHOULD EQUAL Z+1)
C  INPUT :  (I*4)  IL      = NUMBER OF INDEX ENERGY LEVELS
C
C  INPUT :  (I*4)  NDMET   = MAX. NO. OF METASTABLES ALLOWED
C  INPUT :  (I*4)  NDTEM   = MAX. NO. OF TEMPERATURES ALLOWED
C  INPUT :  (I*4)  NDDEN   = MAX. NUMBER OF DENSITIES ALLOWED
C
C  INPUT :  (I*4)  NMET    = NUMBER OF METASTABLES ( 1 -> 5 )
C  INPUT :  (I*4)  IMETR() = INDEX OF METASTABLES IN COMPLETE LEVEL LIST
C
C  INPUT :  (I*4)  IFOUT   = 1 => INPUT TEMPERATURES IN KELVIN
C                            2 => INPUT TEMPERATURES IN EV
C                            3 => INPUT TEMPERATURES IN REDUCED FORM
C  INPUT :  (I*4)  MAXT    = NUMBER OF INPUT TEMPERATURES (1 -> 20)
C  INPUT :  (R*8)  TINE()  = ELECTRON TEMPERATURES (UNITS: SEE 'IFOUT')
C
C  INPUT :  (I*4)  IDOUT   = 1 => INPUT DENSITIES IN CM-3
C                            2 => INPUT DENSITIES IN REDUCED FORM
C  INPUT :  (I*4)  MAXD    = NUMBER OF INPUT DENSITIES (1 -> 20)
C  INPUT :  (R*8)  DINE()  = ELECTRON DENSITIES (UNITS: SEE 'IFOUT')
C
C  INPUT :  (C*18) CSTRGA()= INDEX LEVEL CONFIGURATIONS
C  INPUT :  (R*8)  STCKM(,,)=METASTABLE STATE POPULATIONS:
C                            1ST DIMENSION = METASTABLE STATE INDEX
C                            2ND DIMENSION = TEMPERATURE INDEX
C                            3RD DIMENSION = DENSITY INDEX
C
C           (I*4)  L1      = PARAMETER = 1
C           (I*4)  L2      = PARAMETER = 2
C           (I*4)  L3      = PARAMETER = 3
C
C           (I*4)  I       = GENERAL USE
C           (I*4)  IM      = ARRAY INDEX POINTER FOR METASTABLE STATES
C           (I*4)  IT      = ARRAY INDEX POINTER FOR TEMPERATURES
C           (I*4)  ID      = ARRAY INDEX POINTER FOR DENSITIES
C
C           (R*8)  RDEN()  = ELECTRON DENSITIES (UNITS: REDUCED FORM)
C           (R*8)  RTEM()  = ELECTRON TEMPERATURES (UNITS: REDUCED FORM)
```

```
C
C           (C*1)  CSTAR   = '*'
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          XXTCON     ADAS      CONVERTS ENTERED TEMP. VALUES TO EV.
C          XXDCON     ADAS      CONVERTS ENTERED DENSITY VALUES TO CM-3.
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 4569
C
C DATE:    09/10/90
C
C UPDATE:  20/05/93 - ADAS91 PEB: TO REFLECT CHANGES IN BXDATA THE
C                                 CHARACTER ARRAY CSTRGA IS NOW 18 BYTES
C                                 INSTEAD OF 12.
C                                 NOTE: ONLY THE FIRST 12 BYTES ARE
C                                       OUTPUT TO THE PASSING FILE.
C
C-----------------------------------------------------------------------
      CHARACTER*18        CSTRGA(IL)
      CHARACTER*8         DATE
      INTEGER             IDOUT,       IFOUT,       IL
      INTEGER             IMETR(NMET), IUNIT,       IZ1,         MAXD
      INTEGER             MAXT,        NDDEN,       NDMET,       NDTEM
      INTEGER             NMET
      REAL*8              DINE(MAXD),  STCKM(NDMET,NDTEM,NDDEN)
      REAL*8              TINE(MAXT)
```

3.28 b6ispc: Subroutine b6ispc from library adas2xx

```
      SUBROUTINE B6ISPC( NORD  , IORDR , ISULEV ,
     &                   IORDS
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: B6ISPC *********************
C
C  PURPOSE: TO IDENTIFY IN THE ORDINARY LEVEL INDEX THE INDEX  FOR  THE
C           UPPER LEVEL OF THE SPECIFIC LINE POWER TRANSITION REQUESTED.
C
C  CALLING PROGRAM:  ADAS206
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  NORD    = NUMBER OF ORDINARY EXCITED LEVELS.
C  INPUT : (I*4)  IORDR() = INDEX OF ORDINARY EXCITED LEVELS IN COMPLETE
C                           LEVEL LIST.
C  INPUT : (I*4)  ISULEV  = UPPER ENERGY LEVEL OF SPECIFIC LINE POWER
C                           TRNSITION.
C  OUTPUT: (I*4)  IORDS   = INDEX OF SPECIFIC LINE POWER TRANSITION
C                           UPPER LEVEL IN ORDINARY LEVEL INDEX.
C
C          (I*4)  I4UNIT  = FUNCTION (SEE ROUTINE SECTION BELOW)
C          (I*4)  IO      = ORDINARY EXCITED LEVEL NUMBER COUNTER
C
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          XXTERM     ADAS      TERMINATES PROGRAM WITH MESSAGE
C          I4UNIT     ADAS      FETCH UNIT NUMBER FOR OUTPUT OF MESSAGES
C
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 4569
C
C DATE:    09/10/90
C
C UPDATE:  23/04/93 - PE BRIDEN - ADAS91: ADDED I4UNIT FUNCTION TO WRITE
C                                         STATEMENTS FOR SCREEN MESSAGES
C
C UPDATE:  24/05/93 - PE BRIDEN - ADAS91: CHANGED I4UNIT(0)-> I4UNIT(-1)
C
C UNIX-IDL PORT:
C
C AUTHOR:  WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C
C DATE:    06/06/96
C
C VERSION: 1.1      DATE:06/06/96
C MODIFIED: WILLIAM OSBORN
C     - FIRST VERSION
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IORDR(NORD), IORDS,       ISULEV,      NORD
```

3.29 b6loss: Subroutine b6loss from library adas2xx

```
C
      SUBROUTINE B6LOSS( NDTRN  , NDLEV  ,
     &                   ICNTE  , ISTRN  ,
     &                   XJA    , ER     , AVAL  ,
     &                   IE1A   , IE2A   ,
     &                   SLOSS  , TLOSS
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: B6LOSS *********************
C
C  PURPOSE: TO CALCULATE THE DIRECT LINE POWER LOSS FOR EACH LEVEL AND
C           FOR THE SPECIFIC LINE POWER TRANSITION GIVEN BY 'ISTRN'.
C
C  CALLING PROGRAM:  ADAS206
C
C  SUBROUTINE:
C
C  INPUT :  (I*4)  NDTRN   = MAXIMUM NUMBER OF TRANSITIONS ALLOWED
C  INPUT :  (I*4)  NDLEV   = MAXIMUM NUMBER OF ENERGY LEVELS ALLOWED
C
C  INPUT :  (I*4)  ICNTE   = NUMBER OF ELERCTRON IMPACT TRANSITIONS
C  INPUT :  (I*4)  ISTRN   = SPECIFIC LINE POWER: SELECTED ELECTRON
C                            IMPACT TRANSITION INDEX. (FOR USE WITH
C                            'IE1A()' , 'IE2A()' AND 'AA()' ARRAYS)
C
C
C  INPUT :  (R*8)  XJA()   = QUANTUM NUMBER (J-VALUE) FOR GIVEN LEVEL.
C                            NOTE: (2*XJA)+1 = STATISTICAL WEIGHT
C  INPUT :  (R*8)  ER()    = ENERGY RELATIVE TO LEVEL 1 (RYDBERGS)
C                            DIMENSION: ENERGY LEVEL.
C  INPUT :  (R*8)  AVAL()  = ELECTRON IMPACT TRANSITION: A-VALUE (SEC-1)
C                            DIMENSION: ENERGY LEVEL.
C
C  INPUT :  (I*4)  IE1A()  = ELECTRON IMPACT TRANSITION:
C                            LOWER ENERGY LEVEL INDEX
C  INPUT :  (I*4)  IE2A()  = ELECTRON IMPACT TRANSITION:
C                            UPPER ENERGY LEVEL INDEX
C
C  OUTPUT:  (R*8)  SLOSS    = DIRECT LINE POWER LOSS FOR SPECIFIC LINE
C                             POWER TRANSITION GIVEN BY 'ISTRN'.
C                             (UNITS: ERGS SEC-1)
C  OUTPUT:  (R*8)  TLOSS()  = DIRECT LINE POWER LOSS FOR EACH LEVEL.
C                             (UNITS: ERGS SEC-1)
C                             DIMENSION: LEVEL INDEX
C
C           (R*8)  R2LOSS  = PARAMETER = EQUATION CONSTANT = 2.17958D-11
C                            (CONVERTS RYDBERGS/SEC TO ERGS/SEC)
C
C           (I*4)  LLOWER  = SELECTED ELECTRON IMPACT TRANSITION:
C                            LOWER ENERGY LEVEL INDEX
C           (I*4)  LUPPER  = SELECTED ELECTRON IMPACT TRANSITION:
C                            UPPER ENERGY LEVEL INDEX
C           (I*4)  IC      = TRANSITION ARRAY INDEX
C
C
C ROUTINES:  NONE
C
C NOTES:
```

```
C            EQUATIONS USED -
C
C            FOR EACH TRANSITION - DIRECT LINE POWER LOSS IS GIVEN BY:
C
C            LOSS  = 'R2LOSS' x AVALUE x (ENERGY DIFFERENCE)
C
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/37
C          JET EXT. 5023
C
C DATE:    09/10/90
C
C UPDATE:  29/07/92 - CORRECT ERROR - ZERO TLOSS OVER NDLEV INSTEAD OF
C                                     ICNTE.
C
C UNIX-IDL PORT:
C
C DATE: UNKNOWN
C
C AUTHOR: DAVID H BROOKS, UNIVERSITY OF STRATHCLYDE
C
C*********************************************************************
C PUT UNDER SCCS CONTROL:
C
C DATE:    10-05-96
C
C VERSION: 1.1
C MODIFIED: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C     - FIRST PUT UNDER SCCS
C
C---------------------------------------------------------------------
C
C---------------------------------------------------------------------
      INTEGER             ICNTE,       IE1A(NDTRN), IE2A(NDTRN), ISTRN
      INTEGER             NDLEV,       NDTRN
      REAL*8              AVAL(NDTRN), ER(NDLEV),   SLOSS
      REAL*8              TLOSS(NDLEV),             XJA(NDLEV)
```

3.30 b6lpwr: Subroutine b6lpwr from library adas2xx

```
      SUBROUTINE B6LPWR( NDTEM  ,
     &                   IT     , ICNTE  , IL     , NMET  ,
     &                   IZ1    , IPROJ  ,
     &                            IE1A   , IE2A   , IMETR ,
     &                   TEMP   ,
     &                   ER     , XIA    , EXCRE  ,
     &                   TPL0   , TPLBA
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: B6LPWR *********************
C
C  PURPOSE: TO CALCULATE ZERO DENSITY AND HIGH N PROJECTION LINE POWERS,
C           FOR A GIVEN TEMPERATURE.
C
C  CALLING PROGRAM:  ADAS206
C
C  SUBROUTINE:
C
C  INPUT :  (I*4)  NDTEM   = MAXIMUM NUMBER OF TEMPERATURES ALLOWED
C
C  INPUT :  (I*4)  IT      = INDEX OF TEMPERATURE VALUE BEING ASSESSED
C  INPUT :  (I*4)  ICNTE   = NUMBER OF SELECTED ELECTRON IMPACT TRANSTNS
C  INPUT :  (I*4)  IL      = NUMBER OF ENERGY LEVELS
C  INPUT :  (I*4)  NMET    = NUMBER OF METASTABLE LEVELS
C
C  INPUT :  (I*4)  IZ1     = RECOMBINING ION CHARGE
C  INPUT :  (I*4)  IPROJ   = SPECIFIES INDEX OF LOWEST  LEVEL FOR WHICH
C                            EXTRAPOLATION TO HIGHER N  OF THE RADIATED
C                            POWER IS TO BE PERFORMED. ALL LEVELS ABOVE
C                            AND INCLUDING 'IPROJ' ARE TREATED.
C                           IF 'IPROJ' > 'IL' => EXTRAP'TN SWITCHED OFF
C
C  INPUT :  (I*4)  IE1A()  = ELECTRON IMPACT TRANSITION:
C                            LOWER ENERGY LEVEL INDEX.
C                            DIMENSION: TRANSITION INDEX
C  INPUT :  (I*4)  IE2A()  = ELECTRON IMPACT TRANSITION:
C                            UPPER ENERGY LEVEL INDEX.
C                            DIMENSION: TRANSITION INDEX
C  INPUT :  (I*4)  IMETR() = INDEX OF METASTABLE IN COMPLETE LEVEL LIST
C                            (ARRAY SIZE = 'NDMET' )
C
C  INPUT :  (R*8)  TEMP    = TEMPERATURE (KELVIN)
C
C  INPUT :  (R*8)  ER()    = ENERGY RELATIVE TO LEVEL 1 (RYDBERGS)
C                            DIMENSION: LEVEL INDEX
C  INPUT :  (R*8)  XIA()   = ENERGY RELATIVE TO ION. POT. (RYDBERGS)
C                            DIMENSION: LEVEL INDEX
C  INPUT :  (R*8)  EXCRE(,)= EXCITATION RATE COEFFS (cm**3/s)
C                            1st DIMENSION: TEMPERATURE INDEX
C                            2nd DIMENSION: TRANSITION INDEX
C
C  OUTPUT:  (R*8)  TPL0   = ZERO DENSITY LINE POWER ARISING FROM EXCI-
C                           TATION ONLY FROM THE GROUND LEVEL  FOR  A
C                           GIVEN TEMPERATURE 'TEMP'.
C                           (UNITS: ERGS CM3 SEC-1).
C  OUTPUT:  (R*8)  TPLBA()= HIGH N PROJECTED POWER BASED ON EXCITATIONS
C                           FROM A PARTICULAR METASTABLE TO LEVELS
C                           'IPROJ' UPWARDS FOR A GIVEN TEMPERATURE
```

```
C                          'TEMP'.
C                          (UNITS: ERGS CM3 SEC-1)
C                            DIMENSION: METASTABLE  INDEX
C
C           (R*8)  TK2ATE  = PARAMETER = EQUATION CONSTANT = 1.5789D+05
C           (R*8)  R2LOSS  = PARAMETER = EQUATION CONSTANT = 2.17958D-11
C                            (CONVERTS RYDBERGS/SEC TO ERGS/SEC)
C
C           (I*4)  LLOWER  = SELECTED ELECTRON IMPACT TRANSITION:
C                            LOWER ENERGY LEVEL INDEX
C           (I*4)  LUPPER  = SELECTED ELECTRON IMPACT TRANSITION:
C                            UPPER ENERGY LEVEL INDEX
C           (I*4)  IM      = METASTABLE LEVEL ARRAY INDEX
C           (I*4)  IC      = TRANSITION ARRAY INDEX
C
C           (R*8)  ATE     = EQUATION PARAMETER = 'TK2ATE'/'TEMP'
C           (R*8)  Z1      = 'IZ1'
C           (R*8)  Z2ATE   = 'Z1' * 'Z1' * 'ATE'
C           (R*8)  Z2ATE2  = 1.0 / ('Z1' * 'Z1' * 'ATE')
C           (R*8)  Z2ATEX  = SQRT( 1 / ('Z1' * 'Z1' * 'ATE' * 'ATE' ) )
C           (R*8)  V       = 'Z1' / SQRT('XIA()')
C           (R*8)  VP      = 'V'/(1+'V')
C           (R*8)  ATEL    = 'ATE' * 'XIA(LLOWER)'
C           (R*8)  ATEU    = 'ATE' * 'XIA(LUPPER)'
C           (R*8)  ATEUP   = 'ATEU' * 'VP' * 'VP'
C           (R*8)  PLB1    = USED IN CALCULATING 'PLB'
C           (R*8)  PLB2    = USED IN CALCULATING 'PLB'
C           (R*8)  PLB3    = USED IN CALCULATING 'PLB'
C           (R*8)  PLB     = HIGH N PROJECTED POWER BASED ON EXCITATIONS
C                            FROM A PARTICULAR METASTABLE LEVEL 'LLOWER'
C                            TO  THE  LEVEL  'LUPPER'  FOR  TEMPERATURE
C                            'TEMP'.
C                            (UNITS: ERGS CM3 SEC-1)
C
C ROUTINES: NONE
C
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 4569
C
C DATE:    09/10/90
C
C UPDATE:  24/01/91 - PE BRIDEN: SERIOUS ERROR-'TPLBA()' WAS INCORRECTLY
C                                DECLARED AS INTEGER - IT MUST BE REAL*8
C                                - THEREFORE 'TPLBA()' NOW REAL*8 -
C
C UNIX-IDL PORT:
C
C AUTHOR:  WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C
C DATE:    06/06/96
C
C VERSION: 1.1      DATE:06/06/96
C MODIFIED: WILLIAM OSBORN
C     - FIRST VERSION
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             ICNTE,       IE1A(ICNTE), IE2A(ICNTE), IL
```

```
INTEGER             IMETR(NMET), IPROJ,       IT,          IZ1
INTEGER             NDTEM,       NMET
REAL*8              ER(IL),      EXCRE(NDTEM,ICNTE),       TEMP
REAL*8              TPL0,        TPLBA(NMET), XIA(IL)
```

3.31 b6norm: Subroutine b6norm from library adas2xx

```
      SUBROUTINE B6NORM( NDLEV  , NDMET  ,
     &                   NORD   ,
     &                   STCK   ,
     &                   PLAX   , PLX    ,
     &                   PLASX  , PLSX
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: B6STOT *********************
C
C  PURPOSE: TO NORMALISE TOTAL/SPECIFIC LINE POWERS FOR LEVEL 1
C           AND TOTAL EQUILIBRIUM LINE POWERS TO STAGE TOTAL POPULATION.
C
C  CALLING PROGRAM:  ADAS206
C
C  SUBROUTINE:
C
C  INPUT :  (I*4)  NDLEV   = MAXIMUM NUMBER OF ENERGY LEVELS ALLOWED
C  INPUT :  (I*4)  NDMET   = MAXIMUM NUMBER OF METASTABLE LEVELS ALLOWED
C
C  INPUT :  (I*4)  NORD    = NUMBER OF ORDINARY EXCITED LEVELS
C
C  INPUT :  (R*4)  STCK(,) = POPULATION MATRIX COVERING ALL NON-METAST-
C                            ABLE/ORDINARY EXCITED LEVELS AS FUNCTION
C                            OF METASTABLE INDEX.
C                            VALUES FOR GIVEN TEMPERATURE AND DENSITY.
C                            1st DIMENSION: ORDINARY EXCITED LEVEL INDEX
C                            2nd DIMENSION: METASTABLE LEVEL INDEX
C
C  I/O   :  (R*8)  PLAX    = INPUT:
C                            TOTAL EQUILIBRIUM LINE POWER COEFFICIENTS.
C                            AT FIXED TEMPERATURE AND DENSITY.
C                            (UNITS: ERGS CM3 SEC-1)
C                            OUTPUT:
C                            NORMALISED TO TOTAL STAGE POPULATION
C  I/O   :  (R*8)  PLX     = INPUT:
C                            TOTAL LINE POWERS FOR LEVEL 1 AT FIXED
C                            TEMPERATURE AND DENSITY.
C                            (UNITS: ERGS SEC-1).
C                            OUTPUT:
C                            NORMALISED TO TOTAL STAGE POPULATION
C
C  I/O   :  (R*8)  PLASX   = INPUT:
C                            SPECIFIC EQULIBRIUM LINE PWR COEFFICIENTS.
C                            AT FIXED TEMPERATURE AND DENSITY.
C                            (UNITS: ERGS CM3 SEC-1)
C                            OUTPUT:
C                            NORMALISED TO TOTAL STAGE POPULATION
C  I/O   :  (R*8)  PLSX    = INPUT:
C                            SPECIFIC LINE PWR FOR LEVEL 1 AT FIXED
C                            TEMPERATURE AND DENSITY.
C                            (UNITS: ERGS SEC-1).
C                            OUTPUT:
C                            NORMALISED TO TOTAL STAGE POPULATION
C
C           (I*4)  IS1      = ORDINARY EXCITED LEVEL INDEX
C
C           (R*8)  STOTX    = VARIABLE USED TO SUM STAGE TOTAL POPULATN.
C                             (INITIAL VALUE = 1 => GROUND)
```

```
C
C ROUTINES: NONE
C
C NOTE:
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/37
C          JET EXT. 5023
C
C DATE:    18/05/93
C
C UPDATE:  20/05/93 - P BRIDEN: STCK ARRAY CHANGED FROM REAL*8 -> REAL*4
C
C UNIX-IDL PORT:
C
C AUTHOR:  WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C
C DATE:    06/06/96
C
C VERSION: 1.1      DATE:06/06/96
C MODIFIED: WILLIAM OSBORN
C     - FIRST VERSION
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             NDLEV,       NDMET,       NORD
      REAL*8              PLASX,       PLAX,        PLSX,        PLX
      REAL                STCK(NDLEV,NDMET)
```

3.32 b6spcl: Subroutine b6spcl from library adas2xx

```
      SUBROUTINE B6SPCL( NDLEV   , NDMET   ,
     &                   IORDS   , NMET    ,
     &                   DENSX   ,
     &                   STCKMX  , STACKX ,
     &                   PLAS1   ,
     &                   PLASX   , PLSX
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: B6SPCL *********************
C
C  PURPOSE: TO  CALCULATE  SPECIFIC  LINE  POWERS  FOR  METASTABLES  AND
C           SPECIFIC EQUILIBRIUM LINE POWER.
C
C  CALLING PROGRAM:  ADAS206
C
C  SUBROUTINE:
C
C  INPUT : (I*4) NDLEV    = PARAMETER = MAX NO. OF ENERGY LEVELS ALLOWED
C  INPUT : (I*4) NDMET    = PARAMETER = MAX. NO. OF METASTABLES ALLOWED
C
C  INPUT : (I*4) IORDS    = INDEX OF SPECIFIC LINE POWER TRANSITION
C                           UPPER ENERGY LEVEL IN ORDINARY LEVEL ARRAY.
C  INPUT : (I*4) NMET     = NUMBER OF METASTABLES (1 <= NMET <= 'NDMET')
C
C  INPUT : (R*8) DENSX    = ELECTRON DENSITY (UNITS: CM-3)
C
C  INPUT : (R*8) STCKMX() = METASTABLE POPULATIONS STACK
C                           AT FIXED TEMPERATURE AND DENSITY.
C                            DIMENSION: METASTABLE INDEX
C  INPUT : (R*4) STACKX(,)= ORDINARY EXCITED LEVEL POPULAT'N DEPENDENCE
C                           ON METASTABLE LEVEL. AT FIXED TEMPERATURE
C                           AND DENSITY.
C                             1st DIMENSION: ORDINARY LEVEL INDEX
C                             2nd DIMENSION: METASTABLE INDEX
C
C  INPUT : (R*8)  PLAS1   = DIRECT LINE POWER LOSS FOR SPECIFIC LINE
C                           POWER TRANSITION.
C                           (UNITS: ERGS SEC-1)
C
C  OUTPUT: (R*8)  PLASX   = SPECIFIC EQUILIBRIUM LINE PWR COEFFICIENTS.
C                           AT FIXED TEMPERATURE AND DENSITY.
C                            (UNITS: ERGS CM3 SEC-1)
C  OUTPUT: (R*8)  PLSX()  = SPECIFIC LINE POWERS FOR METASTABLES. THIS
C                           IS THE SUM OF ALL EMISSIONS  ORGINATING IN
C                           THE  COLLISIONAL-RADIATIVE  SENSE FROM THE
C                           METASTABLE. AT FIXED TEMPERATURE AND DENSITY
C                           (UNITS: ERGS SEC-1 )
C                             DIMENSION: METASTABLE  INDEX
C
C          (I*4) IM       = METASTABLE LEVEL ARRAY INDEX
C
C
C
C ROUTINES: NONE
C
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/37
```

```
C          JET EXT. 5023
C
C DATE:    09/10/90
C
C UPDATE:  20/05/93-P BRIDEN: STACKX ARRAY CHANGED FROM REAL*8 -> REAL*4
C
C UNIX-IDL PORT:
C
C AUTHOR:  WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C
C DATE:    06/06/96
C
C VERSION: 1.1      DATE:06/06/96
C MODIFIED: WILLIAM OSBORN
C     - FIRST VERSION
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IORDS,       NDLEV,       NDMET,       NMET
      REAL*8              DENSX,       PLAS1,       PLASX
      REAL*8              PLSX(NDMET)
      REAL                STACKX(NDLEV,NDMET)
      REAL*8              STCKMX(NDMET)
```

3.33 b6totl: Subroutine b6totl from library adas2xx

```
C
      SUBROUTINE B6TOTL( NDLEV   , NDMET  ,
     &                   NORD    , NMET   ,
     &                   IORDR   , IMETR  ,
     &                   DENSX   ,
     &                   STCKMX  , STACKX ,
     &                   PLA1    , PLBAX  ,
     &                   PLAX    , PLX
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: B6TOTL *********************
C
C  PURPOSE: TO  CALCULATE  TOTAL  LINE  POWERS FOR METASTABLES AND TOTAL
C           EQUILIBRIUM LINE POWERS.
C
C  CALLING PROGRAM:  ADAS206
C
C  SUBROUTINE:
C
C  INPUT : (I*4) NDLEV    = PARAMETER = MAX. NO. OF LEVELS ALLOWED
C  INPUT : (I*4) NDMET    = PARAMETER = MAX. NO. OF METASTABLES ALLOWED
C
C  INPUT : (I*4) NORD     = NUMBER OF ORD. LEVELS (1 <= NORD <= 'NDLEV')
C  INPUT : (I*4) NMET     = NUMBER OF METASTABLES (1 <= NMET <= 'NDMET')
C
C  INPUT : (I*4) IORDR()  = INDEX OF ORDINARY LEVELS IN COMPLETE LEVEL
C                           LIST (ARRAY SIZE = 'NDLEV' )
C  INPUT : (I*4) IMETR()  = INDEX OF METASTABLE IN COMPLETE LEVEL LIST
C                           (ARRAY SIZE = 'NDMET' )
C  INPUT : (R*8) DENSX    = ELECTRON DENSITY (UNITS: CM-3)
C
C  INPUT : (R*8) STCKMX() = METASTABLE POPULATIONS STACK
C                           AT FIXED TEMPERATURE AND DENSITY.
C                            DIMENSION: METASTABLE INDEX
C  INPUT : (R*4) STACKX(,)= ORDINARY EXCITED LEVEL POPULAT'N DEPENDENCE
C                           ON METASTABLE LEVEL. AT FIXED TEMPERATURE
C                           AND DENSITY.
C                             1st DIMENSION: ORDINARY LEVEL INDEX
C                             2nd DIMENSION: METASTABLE INDEX
C
C  INPUT : (R*8)  PLA1()  = DIRECT LINE POWER LOSS FOR EACH LEVEL.
C                           (UNITS: ERGS SEC-1) (DIMENSION: LEVEL INDEX)
C  INPUT : (R*8)  PLBAX() = HIGH N PROJECTED POWER BASED ON EXCITATIONS
C                           FROM A PARTICULAR METASTABLE TO LEVELS
C                           'IPROJ' UPWARDS. (UNITS: ERGS CM3 SEC-1)
C                           AT FIXED TEMPERATURE.
C                             DIMENSION: METASTABLE  INDEX
C
C  OUTPUT: (R*8)  PLAX    = TOTAL EQUILIBRIUM LINE POWER COEFFICIENTS.
C                           AT FIXED TEMPERATURE AND DENSITY.
C                            (UNITS: ERGS CM3 SEC-1)
C  OUTPUT: (R*8)  PLX()   = TOTAL LINE POWERS FOR METASTABLES. THIS IS
C                           THE SUM OF ALL EMISSIONS ORGINATING IN THE
C                           COLLISIONAL-RADIATIVE   SENSE   FROM   THE
C                           METASTABLE. AT FIXED TEMPERATURE AND DENSITY
C                           (UNITS: ERGS SEC-1 )
C                             DIMENSION: METASTABLE  INDEX
C
```

```
C          (I*4) IM       = METASTABLE LEVEL ARRAY INDEX
C          (I*4) IS       = ORDINARY LEVEL ARRAY INDEX
C
C
C
C ROUTINES: NONE
C
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/37
C          JET EXT. 5023
C
C DATE:    09/10/90
C
C UPDATE:  20/05/93-P BRIDEN: STACKX ARRAY CHANGED FROM REAL*8 -> REAL*4
C
C UNIX-IDL PORT:
C
C UNIX-IDL PORT:
C
C DATE: UNKNOWN
C
C AUTHOR: DAVID H BROOKS, UNIVERSITY OF STRATHCLYDE
C
C VERSION: 1.1 DATE: 10/05/96
C MODIFIED: WILLIAM OSBORN
C     - FIRST PUT UNDER SCCS
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IMETR(NMET), IORDR(NORD), NDLEV,       NDMET
      INTEGER             NMET,        NORD
      REAL*8              DENSX,       PLA1(NDLEV), PLAX
      REAL*8              PLBAX(NDMET),             PLX(NDMET)
      REAL                STACKX(NDLEV,NDMET)
      REAL*8              STCKMX(NDMET)
```

3.34 b6wr12: Subroutine b6wr12 from library adas2xx

```
      SUBROUTINE B6WR12( IUNIT  , DATE   , IZ1   , IL    ,
     &                   NDMET  , NDTEM  , NDDEN ,
     &                   LNORM  ,
     &                   NMET   , IMETR  ,
     &                   IFOUT  , MAXT   , TINE  ,
     &                   IDOUT  , MAXD   , DINE  ,
     &                   CSTRGA , PL
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: B6WR12 *********************
C
C  PURPOSE: TO OUTPUT TOTAL LINE POWER PARAMETERS TO THE PASSING
C           FILE ON STREAM 'IUNIT'.
C
C  CALLING PROGRAM: ADAS206
C
C  SUBROUTINE:
C
C  INPUT :  (I*4)  IUNIT   = OUTPUT STREAM NUMBER
C  INPUT :  (C*8)  DATE    = CURRENT DATE AS 'DD/MM/YY'
C  INPUT :  (I*4)  IZ1     = RECOMBINING ION CHARGE
C                            (NOTE: IZ1 SHOULD EQUAL Z+1)
C  INPUT :  (I*4)  IL      = NUMBER OF INDEX ENERGY LEVELS
C
C  INPUT :  (I*4)  NDMET   = MAX. NO. OF METASTABLES ALLOWED
C  INPUT :  (I*4)  NDTEM   = MAX. NO. OF TEMPERATURES ALLOWED
C  INPUT :  (I*4)  NDDEN   = MAX. NUMBER OF DENSITIES ALLOWED
C
C  INPUT :  (L*4)  LNORM   =.TRUE.  => IF NMET=1 THEN TOTAL AND SPECIFIC
C                                      LINE POWER OUTPUT FILES PLT/PLS
C                                      NORMALISED TO STAGE TOT.POPULATN.
C                                      (** NORM TYPE = T)
C                          =.FALSE. => OTHERWISE NORMALISE TO IDENTIFIED
C                                      METASTABLE POPULATIONS.
C                                      (** NORM TYPE = M)
C
C  INPUT :  (I*4)  NMET    = NUMBER OF METASTABLES ( 1 -> 5 )
C  INPUT :  (I*4)  IMETR() = INDEX OF METASTABLES IN COMPLETE LEVEL LIST
C
C  INPUT :  (I*4)  IFOUT   = 1 => INPUT TEMPERATURES IN KELVIN
C                            2 => INPUT TEMPERATURES IN EV
C                            2 => INPUT TEMPERATURES IN REDUCED FORM
C  INPUT :  (I*4)  MAXT    = NUMBER OF INPUT TEMPERATURES (1 -> 20)
C  INPUT :  (R*8)  TINE()  = ELECTRON TEMPERATURES (UNITS: SEE 'IFOUT')
C
C  INPUT :  (I*4)  IDOUT   = 1 => INPUT DENSITIES IN CM-3
C                            2 => INPUT DENSITIES IN REDUCED FORM
C  INPUT :  (I*4)  MAXD    = NUMBER OF INPUT DENSITIES (1 -> 20)
C  INPUT :  (R*8)  DINE()  = ELECTRON DENSITIES (UNITS: SEE 'IFOUT')
C
C  INPUT :  (C*18) CSTRGA()= INDEX LEVEL CONFIGURATIONS
C  INPUT :  (R*8)  PL(,,)  = TOTAL LINE POWERS FOR METASTABLES. THIS IS
C                            THE SUM OF ALL EMISSIONS ORGINATING IN THE
C                            COLLISIONAL-RADIATIVE   SENSE   FROM   THE
C                            METASTABLE.
C                             => P(TOTAL)/N(IMET)        (ERGS SEC-1)
C                              1st DIMENSION: METASTABLE  INDEX
C                              2nd DIMENSION: TEMPERATURE INDEX
```

```
C                              3rd DIMENSION: DENSITY     INDEX
C
C           (I*4)  L1       = PARAMETER = 1
C           (I*4)  L2       = PARAMETER = 2
C           (I*4)  L3       = PARAMETER = 3
C
C           (I*4)  I        = GENERAL USE
C           (I*4)  IM       = ARRAY INDEX POINTER FOR METASTABLE STATES
C           (I*4)  IT       = ARRAY INDEX POINTER FOR TEMPERATURES
C           (I*4)  ID       = ARRAY INDEX POINTER FOR DENSITIES
C
C           (R*8)  RDEN()   = ELECTRON DENSITIES (UNITS: REDUCED FORM)
C           (R*8)  RTEM()   = ELECTRON TEMPERATURES (UNITS: REDUCED FORM)
C
C           (C*1)  CSTAR    = '*'
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          XXTCON     ADAS      CONVERTS ENTERED TEMP. VALUES TO EV.
C          XXDCON     ADAS      CONVERTS ENTERED DENSITY VALUES TO CM-3.
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 4569
C
C DATE:    09/10/90
C
C UPDATE:  18/05/93 - PE BRIDEN: ADDED NORMALISATION INFO TO OUTPUT.
C                                NEW ARGUMENT   - LNORM
C                                CHANGED FORMAT - 1011
C
C UPDATE:  20/05/93 - ADAS91 PEB: TO REFLECT CHANGES IN BXDATA THE
C                                  CHARACTER ARRAY CSTRGA IS NOW 18 BYTES
C                                  INSTEAD OF 12.
C                                  NOTE: ONLY THE FIRST 12 BYTES ARE
C                                        OUTPUT TO THE PASSING FILE.
C
C UNIX-IDL PORT:
C
C AUTHOR:  WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C
C DATE:    06/06/96
C
C VERSION: 1.1      DATE:06/06/96
C MODIFIED: WILLIAM OSBORN
C     - FIRST VERSION
C
C-----------------------------------------------------------------------
      CHARACTER*18        CSTRGA(IL)
      CHARACTER*8         DATE
      INTEGER             IDOUT,       IFOUT,       IL
      INTEGER             IMETR(NMET), IUNIT,       IZ1,         MAXD
      INTEGER             MAXT,        NDDEN,       NDMET,       NDTEM
      INTEGER             NMET
      LOGICAL             LNORM
      REAL*8              DINE(MAXD),  PL(NDMET,NDTEM,NDDEN)
      REAL*8              TINE(MAXT)
```

3.35 b6wr13: Subroutine b6wr13 from library adas2xx

```
      SUBROUTINE B6WR13( IUNIT  , DATE   , IZ1    , IL    ,
     &                   NDMET  , NDTEM  , NDDEN  ,
     &                   LNORM  ,
     &                   NMET   , IMETR  ,
     &                   IFOUT  , MAXT   , TINE   ,
     &                   IDOUT  , MAXD   , DINE   ,
     &                   ILOWER , IUPPER ,
     &                   CSTRGA , PLS
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: B6WR13 *********************
C
C  PURPOSE: TO OUTPUT SPECIFIC LINE POWER PARAMETERS TO THE PASSING
C           FILE ON STREAM 'IUNIT'.
C
C  CALLING PROGRAM: ADAS206
C
C  SUBROUTINE:
C
C  INPUT :  (I*4)  IUNIT   = OUTPUT STREAM NUMBER
C  INPUT :  (C*8)  DATE    = CURRENT DATE AS 'DD/MM/YY'
C  INPUT :  (I*4)  IZ1     = RECOMBINING ION CHARGE
C                            (NOTE: IZ1 SHOULD EQUAL Z+1)
C  INPUT :  (I*4)  IL      = NUMBER OF INDEX ENERGY LEVELS
C
C  INPUT :  (I*4)  NDMET   = MAX. NO. OF METASTABLES ALLOWED
C  INPUT :  (I*4)  NDTEM   = MAX. NO. OF TEMPERATURES ALLOWED
C  INPUT :  (I*4)  NDDEN   = MAX. NUMBER OF DENSITIES ALLOWED
C
C  INPUT :  (L*4)  LNORM   =.TRUE.  => IF NMET=1 THEN TOTAL AND SPECIFIC
C                                      LINE POWER OUTPUT FILES PLT/PLS
C                                      NORMALISED TO STAGE TOT.POPULATN.
C                                      (** NORM TYPE = T)
C                          =.FALSE. => OTHERWISE NORMALISE TO IDENTIFIED
C                                      METASTABLE POPULATIONS.
C                                      (** NORM TYPE = M)
C
C  INPUT :  (I*4)  NMET    = NUMBER OF METASTABLES ( 1 -> 5 )
C  INPUT :  (I*4)  IMETR() = INDEX OF METASTABLES IN COMPLETE LEVEL LIST
C
C  INPUT :  (I*4)  IFOUT   = 1 => INPUT TEMPERATURES IN KELVIN
C                            2 => INPUT TEMPERATURES IN EV
C                            2 => INPUT TEMPERATURES IN REDUCED FORM
C  INPUT :  (I*4)  MAXT    = NUMBER OF INPUT TEMPERATURES (1 -> 20)
C  INPUT :  (R*8)  TINE()  = ELECTRON TEMPERATURES (UNITS: SEE 'IFOUT')
C
C  INPUT :  (I*4)  IDOUT   = 1 => INPUT DENSITIES IN CM-3
C                            2 => INPUT DENSITIES IN REDUCED FORM
C  INPUT :  (I*4)  MAXD    = NUMBER OF INPUT DENSITIES (1 -> 20)
C  INPUT :  (R*8)  DINE()  = ELECTRON DENSITIES (UNITS: SEE 'IFOUT')
C
C  INPUT :  (I*4)  ILOWER  = SPECIFIC LINE POWER: SELECTED ELECTRON
C                            IMPACT TRANSITION LOWER LEVEL INDEX
C  INPUT :  (I*4)  IUPPER  = SPECIFIC LINE POWER: SELECTED ELECTRON
C                            IMPACT TRANSITION UPPER LEVEL INDEX
C
C  INPUT :  (C*18) CSTRGA()= INDEX LEVEL CONFIGURATIONS
C  INPUT :  (R*8)  PLS(,,) = SPECIFIC LINE POWERS FOR METASTABLES. THIS
```

```
C                           IS THE SPECIFIC EMISSION ORGINATING IN THE
C                           COLLISIONAL-RADIATIVE   SENSE   FROM   THE
C                           METASTABLE. (SEE 'ISTRN')
C                            => P(SPECIFIC)/N(IMET)  (ERGS SEC-1)
C                              1st DIMENSION: METASTABLE  INDEX
C                              2nd DIMENSION: TEMPERATURE INDEX
C                              3rd DIMENSION: DENSITY     INDEX
C
C           (I*4)  L1      = PARAMETER = 1
C           (I*4)  L2      = PARAMETER = 2
C           (I*4)  L3      = PARAMETER = 3
C
C           (I*4)  I       = GENERAL USE
C           (I*4)  IM      = ARRAY INDEX POINTER FOR METASTABLE STATES
C           (I*4)  IT      = ARRAY INDEX POINTER FOR TEMPERATURES
C           (I*4)  ID      = ARRAY INDEX POINTER FOR DENSITIES
C
C           (R*8)  RDEN()  = ELECTRON DENSITIES (UNITS: REDUCED FORM)
C           (R*8)  RTEM()  = ELECTRON TEMPERATURES (UNITS: REDUCED FORM)
C
C           (C*1)  CSTAR   = '*'
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          XXTCON     ADAS      CONVERTS ENTERED TEMP. VALUES TO EV.
C          XXDCON     ADAS      CONVERTS ENTERED DENSITY VALUES TO CM-3.
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 4569
C
C DATE:    09/10/90
C
C UPDATE:  18/05/93 - PE BRIDEN: ADDED NORMALISATION INFO TO OUTPUT.
C                                NEW ARGUMENT   - LNORM
C                                CHANGED FORMAT - 1011
C
C UPDATE:  20/05/93 - ADAS91 PEB: TO REFLECT CHANGES IN BXDATA THE
C                                  CHARACTER ARRAY CSTRGA IS NOW 18 BYTES
C                                  INSTEAD OF 12.
C                                  NOTE: ONLY THE FIRST 12 BYTES ARE
C                                        OUTPUT TO THE PASSING FILE.
C
C UNIX-IDL PORT:
C
C AUTHOR:  WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C
C DATE:    06/06/96
C
C VERSION: 1.1      DATE:06/06/96
C MODIFIED: WILLIAM OSBORN
C     - FIRST VERSION
C
C-----------------------------------------------------------------------
      CHARACTER*18        CSTRGA(IL)
      CHARACTER*8         DATE
      INTEGER             IDOUT,       IFOUT,       IL,          ILOWER
      INTEGER             IMETR(NMET), IUNIT,       IUPPER,      IZ1
      INTEGER             MAXD,        MAXT,        NDDEN,       NDMET
      INTEGER             NDTEM,       NMET
```

```
LOGICAL             LNORM
REAL*8              DINE(MAXD),  PLS(NDMET,NDTEM,NDDEN)
REAL*8              TINE(MAXT)
```

3.36 b7data: Subroutine b7data from library adas2xx

```
      SUBROUTINE B7DATA( IUNIT  ,
     &                   NDLEV  , NDTEM  , NDDEN  , NDMET ,
     &                   DSNINC , TITLED ,
     &                   IZ     , IZ0    , IZ1    , BWNO  ,
     &                   IL     , NMET   , NORD   ,
     &                   MAXT   , MAXD   , ICNTR  , ICNTH ,
     &                   IA     , ISA    , ILA    , XJA   ,
     &                   CSTRGA ,
     &                   IMETR  , IORDR  , TEA    , DENSA ,
     &                   STCKM  , STVR   , STVH   ,
     &                   STVRM  , STVHM  , STACK
     &                 )
C-----------------------------------------------------------------------
C
C  ******************** FORTRAN77 SUBROUTINE: B7DATA ******************
C
C  PURPOSE:  TO INPUT DATA FROM A CONTOUR PASSING FILE.
C            POPULATION  DATA  FOR DIAGNOSTIC USE.
C
C  CALLING PROGRAM: ADAS207
C
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  IUNIT   = INPUT UNIT NUMBER FOR RESULTS
C
C  INPUT : (I*4)  NDLEV   = MAXIMUM NUMBER OF LEVELS ALLOWED
C  INPUT : (I*4)  NDTEM   = MAXIMUM NUMBER OF TEMPERATURES ALLOWED
C  INPUT : (I*4)  NDDEN   = MAXIMUM NUMBER OF DENSITIES ALLOWED
C  INPUT : (I*4)  NDMET   = MAXIMUM NUMBER OF METASTABLES ALLOWED
C
C  OUTPUT: (C*80) DSNINC  = INPUT COPASE DATA SET NAME (IN QUOTES),
C                           USED TO GENERATE 'CONTOUR' DATA.
C  OUTPUT: (C*3)  TITLED  = ELEMENT SYMBOL.
C
C  OUTPUT: (I*4)  IZ      =  RECOMBINED ION CHARGE
C  OUTPUT: (I*4)  IZ0     =         NUCLEAR CHARGE
C  OUTPUT: (I*4)  IZ1     = RECOMBINING ION CHARGE
C                           (NOTE: IZ1 SHOULD EQUAL IZ+1)
C  OUTPUT: (R*8)  BWNO    = IONISATION POTENTIAL (CM-1)
C
C  OUTPUT: (I*4)  IL      = NUMBER OF ENERGY LEVELS
C  OUTPUT: (I*4)  NMET    = NUMBER OF METASTABLES LEVELS: 1<=NMET<=NDMET
C  OUTPUT: (I*4)  NORD    = NUMBER OF ORDINARY LEVELS ('IL' - 'NMET')
C
C  OUTPUT: (I*4)  MAXT    = NUMBER OF INPUT TEMPERATURES ( 1 -> 'NDTEM')
C  OUTPUT: (I*4)  MAXD    = NUMBER OF INPUT DENSITIES ( 1 -> 'NDDEN')
C  OUTPUT: (I*4)  ICNTR   = NUMBER OF FREE ELECTRON RECOMBINATIONS INPUT
C  OUTPUT: (I*4)  ICNTH   = NO. OF CHARGE EXCHANGE RECOMBINATIONS INPUT
C
C  OUTPUT: (I*4)  IA()    = ENERGY LEVEL INDEX NUMBER
C  OUTPUT: (I*4)  ISA()   = MULTIPLICITY FOR LEVEL 'IA()'
C                           NOTE: (ISA-1)/2 = QUANTUM NUMBER (S)
C  OUTPUT: (I*4)  ILA()   = QUANTUM NUMBER (L) FOR LEVEL 'IA()'
C  OUTPUT: (R*8)  XJA()   = QUANTUM NUMBER FOR LEVEL 'IA()'
C                           NOTE: (2*XJA)+1 = STATISTICAL WEIGHT
C
C  OUTPUT: (C*18) CSTRGA()= NOMENCLATURE/CONFIGURATION FOR LEVEL 'IA()'
C
```

```
C  OUTPUT: (I*4)  IMETR() = INDEX OF METASTABLE IN COMPLETE LEVEL LIST
C  OUTPUT: (I*4)  IORDR() = INDEX OF ORDINARY LEVELS IN COMPLETE LEVEL
C                           LIST.
C  OUTPUT: (R*8)  TEA()   = ELECTRON TEMPERATURES (UNITS: KELVIN)
C  OUTPUT: (R*8)  DENSA() = ELECTRON DENSITIES  (UNITS: CM-3)
C
C  OUTPUT: (R*8) STCKM(,,) = METASTABLE POPULATIONS STACK
C                            1st DIMENSION: METASTABLE INDEX
C                            2nd DIMENSION: TEMPERATURE INDEX
C                            3rd DIMENSION: DENSITY INDEX
C  OUTPUT: (R*8) STVR(,,)  = FREE ELECTRON RECOMBINATION COEFFICIENTS
C                            1st DIMENSION: LEVEL INDEX
C                            2nd DIMENSION: TEMPERATURE INDEX
C                            3rd DIMENSION: DENSITY INDEX
C  OUTPUT: (R*8) STVH(,,)  =  CHARGE EXCHANGE COEFFICIENTS
C                            1st DIMENSION: LEVEL INDEX
C                            2nd DIMENSION: TEMPERATURE INDEX
C                            3rd DIMENSION: DENSITY INDEX
C  OUTPUT: (R*8) STVRM(,,) = METASTABLE FREE ELECTRON RECOMBINATION
C                           COEFFICIENTS.
C                            1st DIMENSION: METASTABLE INDEX
C                            2nd DIMENSION: TEMPERATURE INDEX
C                            3rd DIMENSION: DENSITY INDEX
C  OUTPUT: (R*8) STVHM(,,) = METASTABLE CHARGE EXCHANGE COEFFICIENTS
C                            1st DIMENSION: METASTABLE INDEX
C                            2nd DIMENSION: TEMPERATURE INDEX
C                            3rd DIMENSION: DENSITY INDEX
C  OUTPUT: (R*8) STACK(,,,)= POPULATION DEPENDENCE
C                            1st DIMENSION: LEVEL INDEX
C                            2nd DIMENSION: METASTABLE INDEX
C                            3rd DIMENSION: TEMPERATURE INDEX
C                            4th DIMENSION: DENSITY INDEX
C
C          (I*4) I4UNIT    = FUNCTION (SEE ROUTINE SECTION BELOW)
C          (I*4) I         = GENERAL USE
C          (I*4) J         = GENERAL USE
C          (I*4) K         = GENERAL USE
C          (I*4) L         = GENERAL USE
C
C NOTE:
C          THIS INPUT DATA IS FROM THE PROGRAM 'SPFPOPN/P'
C
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4UNIT     ADAS      FETCH UNIT NUMBER FOR OUTPUT OF MESSAGES
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/37
C          JET EXT. 5023
C
C DATE:    09/10/90
C
C UPDATE:  22/10/92 - PEB: INCLUDED ERROR HANDLING FOR ARRAY OVERFLOWS
C
C UPDATE:  23/04/93 - PE BRIDEN - ADAS91: ADDED I4UNIT FUNCTION TO WRITE
C                                         STATEMENTS FOR SCREEN MESSAGES
C
C UPDATE:  24/05/93 - PE BRIDEN - ADAS91: CHANGED I4UNIT(0)-> I4UNIT(-1)
C
```

```
C UPDATE:  28/01/94 - PE BRIDEN - ADAS91: INCREASED CSTRGA C*12 -> C*18
C                                         FORMAT 1003 CHANGED ACCORDINGLY
C
C UPDATE:  09/03/95 - SP BELLAMY -  UNIX: INCREASE DSNINC TO 80
C                                          AND CHANGE FORMAT 1000
C
C-----------------------------------------------------------------------
      CHARACTER*18        CSTRGA(NDLEV)
      CHARACTER*80        DSNINC
      CHARACTER*3         TITLED
      INTEGER             IA(NDLEV),   ICNTH,       ICNTR,       IL
      INTEGER             ILA(NDLEV),  IMETR(NDMET)
      INTEGER             IORDR(NDLEV),             ISA(NDLEV),  IUNIT
      INTEGER             IZ,          IZ0,         IZ1,         MAXD
      INTEGER             MAXT,        NDDEN,       NDLEV,       NDMET
      INTEGER             NDTEM,       NMET,        NORD
      REAL*8              BWNO,        DENSA(NDDEN)
      REAL*8              STACK(NDLEV,NDMET,NDTEM,NDDEN)
      REAL*8              STCKM(NDMET,NDTEM,NDDEN)
      REAL*8              STVH(NDLEV,NDTEM,NDDEN)
      REAL*8              STVHM(NDMET,NDTEM,NDDEN)
      REAL*8              STVR(NDLEV,NDTEM,NDDEN)
      REAL*8              STVRM(NDMET,NDTEM,NDDEN), TEA(NDTEM)
      REAL*8              XJA(NDLEV)
```

3.37 b7datc: Subroutine b7datc from library adas2xx

```
       SUBROUTINE B7DATC( ndlev   ,
     &                    titled_c , iz_c     , iz0_c  , iz1_c ,
     &                    bwno_c   , il_c     ,
     &                    ia_c     , cstrga_c , isa_c  , ila_c , xja_c ,
     &                    titled   , iz       , iz0    , iz1   ,
     &                    bwno     , il       ,
     &                    ia       , cstrga   , isa    , ila   , xja
     &                  )

C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: B7DATC *********************
C
C  PURPOSE:  Check consistency of adf04 file with contour passing file
C            data.
C
C  CALLING PROGRAM: ADAS207
C
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  ndlev     = maximum number of levels that can be read
C
C  INPUT : (C*3)  titled_c  = contour file: element symbol.
C  INPUT : (I*4)  iz_c      = contour file: recombined ion charge read
C  INPUT : (I*4)  iz0_c     = contour file: nuclear charge read
C  INPUT : (I*4)  iz1_c     = contour file: recombining ion charge read
C  INPUT : (R*8)  bwno_c    = contour file: ionisation potential (cm-1)
C  INPUT : (I*4)  il_c      = contour file: number of energy levels
C  INPUT : (I*4)  ia_c()    = contour file: energy level index number
C  INPUT : (C*18) cstrga_c()= contour file: configuration for level 'ia()'
C  INPUT : (I*4)  isa_c()   = contour file: multiplicity for level 'ia()'
C  INPUT : (I*4)  ila_c()   = contour file: quantum no. (L) for level 'ia()'
C  INPUT : (R*8)  xja_c()   = contour file: quantum no.(J) for level 'ia()'
C
C  INPUT : (C*3)  titled    = adf04 file: element symbol.
C  INPUT : (I*4)  iz        = adf04 file: recombined ion charge read
C  INPUT : (I*4)  iz0       = adf04 file: nuclear charge read
C  INPUT : (I*4)  iz1       = adf04 file: recombining ion charge read
C  INPUT : (R*8)  bwno      = adf04 file: ionisation potential (cm-1)
C  INPUT : (I*4)  il        = adf04 file: number of energy levels
C  INPUT : (I*4)  ia()      = adf04 file: energy level index number
C  INPUT : (C*18) cstrga()  = adf04 file: configuration for level 'ia()'
C  INPUT : (I*4)  isa()     = adf04 file: multiplicity for level 'ia()'
C  INPUT : (I*4)  ila()     = adf04 file: quantum no. (L) for level 'ia()'
C  INPUT : (R*8)  xja()     = adf04 file: quantum no.(J) for level 'ia()'
C
C
C NOTE: Replaces original b7datc.for by Paul Briden. This version only
C       checks consistency and returns the transition energy.
C
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4UNIT     ADAS      FETCH UNIT NUMBER FOR OUTPUT OF MESSAGES
C
C
C VERSION: 1.7                          DATE: 01-05-2003
```

```
C MODIFIED: Martin O'Mullane
C     - Rewrite and functionality changed.
C
C-----------------------------------------------------------------------
      CHARACTER*18        CSTRGA(NDLEV),            CSTRGA_C(NDLEV)
      CHARACTER*3         TITLED,      TITLED_C
      INTEGER             IA(NDLEV),   IA_C(NDLEV), IL
      INTEGER             ILA(NDLEV),  ILA_C(NDLEV),             IL_C
      INTEGER             ISA(NDLEV),  ISA_C(NDLEV),             IZ
      INTEGER             IZ0,         IZ0_C,       IZ1,         IZ1_C
      INTEGER             IZ_C,        NDLEV
      REAL*8              BWNO,        BWNO_C,      XJA(NDLEV)
      REAL*8              XJA_C(NDLEV)
```

3.38 b7lrat: Subroutine b7lrat from library adas2xx

```
      SUBROUTINE B7LRAT( EM1    , EM2    ,
     &                   RMIN   , RMAX   ,
     &                   RAT
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: B7LRAT *********************
C
C  PURPOSE: TO CALCULATE THE SPECTRUM-LINE INTENSITY RATIO FOR TWO
C           COMPOSITE LINES FROM THEIR INDIVIDUAL SPECTRUM LINE
C           INTENSITIES.
C
C           INTENSITIES AT FIXED TEMPERATURE AND DENSITY. 'RMIN'
C           AND 'RMAX' CONTAIN MINIMUM AND MAXIMUM RATIO VALUES.
C
C  CALLING PROGRAM: ADAS207
C
C  SUBROUTINE:
C
C  INPUT : (R*8)  EM1     = FIRST COMPOSITE ASSEMBLY SPECTRUM-LINE
C                           INTENSITY, AT FIXED TEMPERATURE & DENSITY.
C  INPUT : (R*8)  EM2     = SECOND COMPOSITE ASSEMBLY SPECTRUM-LINE
C                           INTENSITY, AT FIXED TEMPERATURE & DENSITY.
C
C  I/O   : (R*8)  RMIN    = MINIMUM SPECTRUM-LINE INTENSITY RATIO VALUE
C  I/O   : (R*8)  RMAX    = MAXIMUM SPECTRUM-LINE INTENSITY RATIO VALUE
C
C  OUTPUT: (R*8)  RAT     = SPECTRUM-LINE INTENSITY RATIO
C
C
C ROUTINES: NONE
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 4569
C
C DATE:    09/10/90
C
C VERSION: 1.2
C DATE: 20-07-07
C MODIFIED: Allan Whiteford
C           - Small modification to comments to allow for automatic
C             documentation preparation.
C
C-----------------------------------------------------------------------
      REAL*8              EM1,         EM2,         RAT,         RMAX
      REAL*8              RMIN
```

3.39 b8corp: Subroutine b8corp from library adas2xx

```
C
      SUBROUTINE B8CORP(NLEV   , MAXT   , MAXD   , NMET  ,
     &                  TEVA   , COEFF  )


C-----------------------------------------------------------------------
C
C  ******************* FORTRAN77 SUBROUTINE: B8CORP ********************
C
C  PURPOSE: Corrects unphysical low temperature recombination
C           contributions to a PEC.
C
C           There is a low temperature problem in the production
C           calculation which gives unphysical recombination
C           contributions to the PECs. Generally the first 3-4
C           temperatures in the ADAS 96 standard are affected. This
C           routine replaces the first 4 temperatures from the
C           recombination contribution with extrapolated values
C           from the remaining data.
C
C
C  CALLING PROGRAM: ADAS208 (B8WR11)
C
C
C  INPUT : (I*4)  NLEV    = NUMBER OF LEVELS
C  INPUT : (I*4)  MAXT    = NUMBER OF TEMPERATURES
C  INPUT : (I*4)  MAXD    = NUMBER OF DENSITIES
C  INPUT : (I*4)  NMET    = NUMBER OF METASTABLES
C
C  I/O   : (R*4) COEFF()  = RECOMBINATION CONTRIBUTION TO THE PEC
C                           (STVR IN ADAS208 CALL)
C                           1ST DIMENSION : LEVELS
C                           2ND DIMENSION : TEMPERATURES
C                           3RD DIMENSION : DENSITIES
C                           4TH DIMENSION : METASTABLES
C
C
C ROUTINES:
C           ROUTINE    SOURCE    BRIEF DESCRIPTION
C           ------------------------------------------------------------
C           XXSPLN     ADAS      SPLINE SUBROUTINE
C           R8FUN1     ADAS      REAL*8 FUNCTION: ( X -> X )
C
C AUTHOR:  Martin O'Mullane
C
C
C DATE:    14-09-99
C
C
C VERSION: 1.1                          DATE: 14-09-99
C MODIFIED: Martin O'Mullane
C           - First version
C
C VERSION : 1.2
C DATE    : 26-10-99
C MODIFIED: Martin O'Mullane
C           - Consider each level separately for extrapolation.
C           - Increase NDLEV in line with adas208.
C
```

```
C VERSION : 1.3
C DATE    : 20-07-07
C MODIFIED: Allan Whiteford
C           - Small modification to comments to allow for automatic
C             documentation preparation.
C
C VERSION : 1.4
C DATE    : 02-09-2007
C MODIFIED: Martin O'Mullane
C           - ITAG was defined as real*8 rather than integer.
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             MAXD,        MAXT,        NLEV,        NMET
      REAL                COEFF(NDLEV,NDTEM,NDDEN,NDMET)
      REAL*8              TEVA(NDTEM)
```

3.40 b8corr: Subroutine b8corr from library adas2xx

```
C
      SUBROUTINE B8CORR(MAXT, MAXD, NMET, NPL3, TEVA, COEFF)



C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: B8CORR *********************
C
C  PURPOSE: Corrects unphysical low temperature recombination
C           contributions to a PEC.
C
C           There is a low temperature problem in the production
C           calculation which gives unphysical recombination
C           coefficients. Generally the first 3-4 temperatures in the
C           ADAS 96 standard are affected. This routine replaces the
C           first 4 temperatures from the  recombination contribution
C           with extrapolated values from the remaining data.
C
C
C
C  CALLING PROGRAM: ADAS208 (B8WRMC)
C
C  INPUT : (I*4)  MAXT    = NUMBER OF TEMPERATURES
C  INPUT : (I*4)  MAXD    = NUMBER OF DENSITIES
C  INPUT : (I*4)  NMET    = NUMBER OF METASTABLES
C  INPUT : (I*4)  NPL3    = NUMBER OF ACTIVE METAS. FOR RE+3B OF (Z+1) ION
C
C  I/O   : (R*8) COEFF()  = (Z+1)-(Z) RECOM GEN. COLL. RAD. COEFFTS.
C                           (FVRRED IN ADAS208 CALL)
C                             1ST DIMENSION: (Z) METASTABLE INDEX
C                             2ND DIMENSION: (Z) METASTABLE INDEX
C                             3RD DIMENSION: TEMPERATURE INDEX
C                             4TH DIMENSION: DENSITY INDEX
C
C
C
C ROUTINES:
C           ROUTINE    SOURCE    BRIEF DESCRIPTION
C           ------------------------------------------------------------
C           XXSPLE     ADAS      SPLINE SUBROUTINE
C           R8FUN1     ADAS      REAL*8 FUNCTION: ( X -> X )
C
C AUTHOR:  Martin O'Mullane
C
C
C DATE:    14-09-99
C
C
C VERSION: 1.1                          DATE: 14-09-99
C MODIFIED: Martin O'Mullane
C           - First version
C
C VERSION: 1.2                          DATE: 26-10-99
C MODIFIED: Martin O'Mullane
C           - Change the condition for extrapolating. If there are
C             coeff .GT. 1.0 then initiate extrapolation. Also
C             consider eachmetastable separately.
C
```

```
C VERSION: 1.3                          DATE: 20-07-07
C MODIFIED: Allan Whiteford
C           - Small modification to comments to allow for automatic
C             documentation preparation.
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             MAXD,        MAXT,        NMET,        NPL3
      REAL*8              COEFF(NDMET,NDMET,NDTEM,NDDEN)
      REAL*8              TEVA(NDTEM)
```

3.41 b8cors: Subroutine b8cors from library adas2xx

```
      SUBROUTINE B8CORS(NUMT , TEVA , COEFF)


C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: B8CORP *******************
C
C  PURPOSE: There is a low temperature problem in the calculation
C           of the sxb coefficients. Replace the effetive zero
C           values with an exp(E/kT) derived from first good points.
C
C
C  CALLING PROGRAM: ADAS208 (B8WRPS)
C
C
C  INPUT : (I*4)  NUMT    = NUMBER OF TEMPERATURES
C  INPUT : (I*4)  TEVA    = TEMPERATURES
C
C  I/O   : (R*4) COEFF()  = SXB DATA
C
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          XXSPLN     ADAS      SPLINE SUBROUTINE
C          R8FUN1     ADAS      REAL*8 FUNCTION: ( X -> X )
C
C AUTHOR:  Martin O'Mullane
C
C
C DATE:    17-02-2006
C
C
C VERSION : 1.1
C DATE    : 17-02-2006
C MODIFIED: Martin O'Mullane
C           - First version
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             NUMT
      REAL*8              COEFF(NUMT), TEVA(NUMT)
```

3.42 b8data: Subroutine b8data from library adas2xx

```
C
       SUBROUTINE B8DATA( IUNIT  , NDLEV  , NDTRN , NDMET ,
     &                    TITLED , IZ     , IZ0   , IZ1   , BWNO  ,
     &                    NPL    , BWNOA  , LBSETA, PRTWTA, CPRTA ,
     &                    IL     ,
     &                    IA     , CSTRGA , ISA   , ILA   , XJA   , WA ,
     &                    CPLA   , NPLA   , IPLA  , ZPLA  ,
     &                    NV     , SCEF   ,
     &                    ITRAN  , MAXLEV ,
     &                    TCODE  , I1A    , I2A   , AVAL  , SCOM
     &                  )
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: B8DATA *********************
C
C  PURPOSE:  TO FETCH DATA FROM INPUT COPASE DATA SET, INCLUDING
C            MULTIPLE PARENTS ON FREE-ELECTRON AND CHARGE EXCHANGE
C            ON RECOMBINATION, INCLUSION OF EXPLICIT CONTRIBUTIONS BY
C            IONISATION.
C
C            IMPROVEMENT OF AUTOMATIC IONISATION CALC. BY INCLUDING
C            ASSIGNMENT OF FINAL STATE PARENT.
C
C            This is a 208 specfic version of badata which recognizes
C            S lines in the adf04 file.
C
C  CALLING PROGRAM: ADAS208
C
C  DATA:
C           THE 'REAL' DATA IN THE FILE IS REPRESENTED IN AN ABBREVIATED
C           FORM WHICH OMITS THE "D" OR "E" EXPONENT SPECIFIER.
C           e.g. 1.23D-06 or 1.23E-06 IS REPRESENTED AS 1.23-06
C                6.75D+07 or 6.75E+07 IS REPRESENTED AS 6.75+07
C
C           THEREFORE THE FORM OF EACH 'REAL' NUMBER IN THE DATA SET IS:
C                          N.NN+NN or N.NN-NN
C
C           THE UNITS USED IN THE DATA FILE ARE TAKEN AS FOLLOWS:
C
C           IONISATION POTENTIAL: WAVE NUMBER (CM-1)
C           INDEX LEVEL ENERGIES: WAVE NUMBER (CM-1)
C           TEMPERATURES        : KELVIN
C           A-VALUES            : SEC-1
C           GAMMA-VALUES        :
C           RATE COEFFT.        : CM3 SEC-1
C
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  IUNIT   = UNIT TO WHICH INPUT FILE IS ALLOCATED
C  INPUT : (I*4)  NDLEV   = MAXIMUM NUMBER OF LEVELS THAT CAN BE READ
C  INPUT : (I*4)  NDTRN   = MAX. NUMBER OF TRANSITIONS THAT CAN BE READ
C  INPUT : (I*4)  NDMET   = MAX. NUMBER OF METASTABLES ALLOWED
C
C  OUTPUT: (C*3)  TITLED  = ELEMENT SYMBOL.
C  OUTPUT: (I*4)  IZ      =  RECOMBINED ION CHARGE READ
C  OUTPUT: (I*4)  IZ0     =         NUCLEAR CHARGE READ
C  OUTPUT: (I*4)  IZ1     = RECOMBINING ION CHARGE READ
```

```
C                              (NOTE: IZ1 SHOULD EQUAL IZ+1)
C  OUTPUT: (R*8)  BWNO    = IONISATION POTENTIAL (CM-1) OF LOWEST PARENT
C  OUTPUT: (I*4)  NPL     = NUMBER OF PARENTS ON FIRST LINE AND USED
C                           IN LEVEL ASSIGNMENTS
C  OUTPUT: (R*8)  BWNOA() = IONISATION POTENTIAL (CM-1) OF PARENTS
C  OUTPUT: (L*4)  LBSETA()= .TRUE.  - PARENT WEIGHT SET FOR BWNOA()
C                           .FALSE. - PARENT WEIGHT NOT SET FOR BWNOA()
C  OUTPUT: (R*8)  PRTWTA()= PARENT WEIGHT FOR BWNOA()
C  OUTPUT: (C*9)  CPRTA() = PARENT NAME IN BRACKETS
C
C  OUTPUT: (I*4)  IL      = INPUT DATA FILE: NUMBER OF ENERGY LEVELS
C
C  OUTPUT: (I*4)  IA()    = ENERGY LEVEL INDEX NUMBER
C  OUTPUT: (C*18) CSTRGA()= NOMENCLATURE/CONFIGURATION FOR LEVEL 'IA()'
C  OUTPUT: (I*4)  ISA()   = MULTIPLICITY FOR LEVEL 'IA()'
C                           NOTE: (ISA-1)/2 = QUANTUM NUMBER (S)
C  OUTPUT: (I*4)  ILA()   = QUANTUM NUMBER (L) FOR LEVEL 'IA()'
C  OUTPUT: (R*8)  XJA()   = QUANTUM NUMBER (J-VALUE) FOR LEVEL 'IA()'
C                           NOTE: (2*XJA)+1 = STATISTICAL WEIGHT
C  OUTPUT: (R*8)  WA()    = ENERGY RELATIVE TO LEVEL 1 (CM-1) FOR LEVEL
C                           'IA()'
C  OUTPUT: (C*1)  CPLA()  = CHAR. SPECIFYING 1ST PARENT FOR LEVEL 'IA()'
C                                INTEGER - PARENT IN BWNOA() LIST
C                                'BLANK' - PARENT BWNOA(1)
C                                  'X'   - DO NOT ASSIGN A PARENT
C  OUTPUT: (I*4)  NPLA()  = NO. OF PARENT/ZETA CONTRIBUTIONS TO IONIS.
C                           OF LEVEL
C  OUTPUT: (I*4)  IPLA(,) = PARENT INDEX FOR CONTRIBUTIONS TO IONIS.
C                           OF LEVEL
C                           1ST DIMENSION: PARENT INDEX
C                           2ND DIMENSION: LEVEL INDEX
C  OUTPUT: (I*4)  ZPLA(,) = EFF. ZETA PARAM. FOR CONTRIBUTIONS TO IONIS.
C                           OF LEVEL
C                           1ST DIMENSION: PARENT INDEX
C                           2ND DIMENSION: LEVEL INDEX
C
C  OUTPUT: (I*4)  NV      = INPUT DATA FILE: NUMBER OF GAMMA/TEMPERATURE
C                           PAIRS FOR A GIVEN TRANSITION.
C  OUTPUT: (R*8)  SCEF()  = INPUT DATA FILE: ELECTRON TEMPERATURES (K)
C                           (INITIALLY JUST THE MANTISSA. SEE 'ITPOW()')
C                           (NOTE: TE=TP=TH IS ASSUMED)
C
C  OUTPUT: (I*4)  ITRAN   = INPUT DATA FILE: NUMBER OF TRANSITIONS
C  OUTPUT: (I*4)  MAXLEV  = HIGHEST INDEX LEVEL IN READ TRANSITIONS
C
C  OUTPUT: (C*1)  TCODE() = TRANSITION: DATA TYPE POINTER:
C                           ' ' => Electron Impact   Transition
C                           'P' => Proton   Impact   Transition
C                           'H' => Charge   Exchange Recombination
C                           'R' => Free     Electron Recombination
C                           'I' => Coll. ionisation from lower stage ion
C                           'S' => Ionisation
C  OUTPUT: (I*4)  I1A()   = TRANSITION:
C                            LOWER ENERGY LEVEL INDEX (CASE ' ' & 'P')
C                            SIGNED PARENT NDEX (CASE 'H','R','S' & 'I')
C  OUTPUT: (I*4)  I2A()   = TRANSITION:
C                            UPPER ENERGY LEVEL INDEX (CASE ' ' & 'P')
C                            CAPTURING LEVEL INDEX (CASE 'H','R' ,'S'& 'I')
C  OUTPUT: (R*8)  AVAL()  = TRANSITION:
C                            A-VALUE (SEC-1)          (CASE ' ')
C                            NEUTRAL BEAM ENERGY      (CASE 'H')
```

```
C                              NOT USED             (CASE 'P','R','S' & 'I')
C  OUTPUT: (R*8)  SCOM(,) = TRANSITION:
C                              GAMMA VALUES             (CASE ' ' & 'P')
C                              RATE COEFFT.(CM3 SEC-1)(CASE 'H','R' & 'I')
C                              SCALED RATE COEFFT.(CM3 SEC-1)(CASE 'S')
C                             1ST DIMENSION - TEMPERATURE 'SCEF()'
C                             2ND DIMENSION - TRANSITION NUMBER
C
C          (I*4)  NVMAX   = PARAMETER = MAX. NUMBER OF TEMPERATURES
C                                       THAT CAN BE READ IN.
C          (I*4)  MTIED   = PARAMETER = MUST BE GREATER THAN OR EQUAL TO
C                                       THE MAX. NO. OF LEVELS.
C          (R*8)  DZERO   = PARAMETER = MINIMUM VALUE FOR 'AVAL()' AND
C                                       'SCOM()' ARRAYS = 1.0D-30
C
C          (I*4)  I4UNIT  = FUNCTION (SEE ROUTINE SELECTION BELOW)
C          (I*4)  IQS     = X-SECT DATA FORMAT SELECTOR
C                           NOTE: IQS=3 ONLY ALLOWED IN THIS PROGRAM
C          (I*4)  IFAIL   = FAILURE NUMBER FROM B9PARS AND B9PRS1
C          (I*4)  I       = GENERAL USE.
C          (I*4)  IABT    = RETURN CODE FROM 'R(FCTN' (0 => NO ERROR)
C                           OR FROM INTERROGATION OF 'C7'
C          (I*4)  J       = GENERAL USE.
C          (I*4)  J1      = INPUT DATA FILE - SELECTED TRANSITION:
C                            LOWER ENERGY LEVEL INDEX (CASE ' ' & 'P')
C          (I*4)  J2      = INPUT DATA FILE - SELECTED TRANSITION:
C                            UPPER ENERGY LEVEL INDEX (CASE ' ' & 'P')
C                            CAPTURING    LEVEL INDEX (CASE 'H' & 'R')
C          (I*4)  LENCST  = BYTE LENGTH OF STRING CSTRGA()
C          (I*4)  ILINE   = ENERGY LEVEL INDEX FOR CURRENT LINE
C          (I*4)  IRECL   = RECORD LENGTH OF INPUT DATASET (<=128)
C          (I*4)  IAPOW   = EXPONENT OF 'AVALM'
C          (I*4)  IGPOW() = EXPONENT OF 'GAMMA()'
C          (I*4)  ITPOW() = TEMPERATURES - EXPONENT
C                           NOTE: MANTISSA INITIALLY KEPT IN 'SCEF()'
C
C          (R*4)  ZF      = SHOULD BE EQUIVALENT TO 'IZ1'
C
C          (R*8)  AVALM   = INPUT DATA FILE - SELECTED TRANSITION:
C                           MANTISSA OF:   ('IAPOW' => EXPONENT)
C                            A-VALUE (SEC-1)          (CASE ' ')
C                            NEUTRAL BEAM ENERGY      (CASE 'H')
C                            NOT USED              (CASE 'P','R','S' & 'I')
C          (R*8)  GAMMA() = INPUT DATA FILE - SELECTED TRANSITION:
C                           MANTISSA OF: ('IGPOW()' => EXPONENT)
C                            GAMMA VALUES             (CASE ' ' & 'P')
C                            RATE COEFFT.(CM3 SEC-1)(CASE 'H','R','S' & 'I')
C                           DIMENSION => TEMPERATURE 'SCEF()'
C
C          (C*7)  C7      = USED TO PARSE VALUE FOR XJA()
C          (C*7)  CDELIM  = DELIMITERS FOR INPUT OF DATA FROM HEADERS
C          (C*18) C18     = USED TO PARSE VALUE TO CSTRGA()
C          (C*18) C18T    = COPY OF C18 : UNSATISFACTORY METHOD OF
C                           AVOIDING COMPILER REFERENCE ERROR :
C                           DHB 07.04.95
C          (C*80) CLINE   = CURRENT ENERGY LEVEL INDEX PARAMETER LINE
C          (C*75) STRING  = TAIL STRING OF 1ST DATA LINE FOR PARSING
C          (C*44) STRG1   = TAIL STRING OF LEVEL SPEC LINES FOR PARSING
C          (C*128)BUFFER  = GENERAL STRING BUFFER STORAGE
C          (C*3)  CITPOW()= USED TO PARSE VALUES TO ITPOW()
C          (C*5)  CSCEF() = USED TO PARSE VALUES TO SCEF()
```

```
C
C          (L*4)  LDATA   = IDENTIFIES  WHETHER  THE END OF AN  INPUT
C                           SECTION IN THE DATA SET HAS BEEN LOCATED.
C                           (.TRUE. => END OF SECTION REACHED)
C          (L*4)  LTCHR   = .TRUE.  => CURRENT 'TCODE()' = 'H' OR 'R'
C                                                        'S' OR 'I'
C                         = .FALSE. => CURRENT 'TCODE()'.NE.'H' OR 'R'
C                                                          'S' OR 'I'
C          (L*4)  LTCPR   = .TRUE.  => CURRENT 'TCODE()' = 'P' OR 'R'
C                                                             OR 'I'
C                         = .FALSE. => CURRENT 'TCODE()'.NE. 'P' OR 'R'
C                                                           'S' OR 'I'
C          (L*4)  LERROR  = .TRUE.  => UNTIED LEVEL FOUND
C                         = .FALSE. => ALL LEVELS TIED
C          (L*4)  LTIED() = .TRUE.  => SPECIFIED LEVEL TIED
C                         = .FALSE. => SPECIFIED LEVEL IS UNTIED
C                           DIMENSION => LEVEL INDEX
C
C
C NOTE:             LTCHR        LTCPR         TCODE()
C                  ----------------------------------
C                  .TRUE.       .TRUE.    =>     'R','I'
C                  .TRUE.       .FALSE.   =>     'H'
C                  .FALSE.      .TRUE.    =>     'P'
C                  .FALSE.      .FALSE.   =>     ' '
C
C        FOR A-VALUES & GAMMA-VALUES ENTRIES LESS THAN 'DZERO' ARE TAKEN
C        AS BEING EQUAL TO DZERO. THIS AFFECTS THE 'AVAL()' AND 'SCOM()'
C        ARRAYS.
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4UNIT     ADAS      FETCH UNIT NUMBER FOR OUTPUT OF MESSAGES
C          R8FCTN     ADAS      CONVERTS FROM CHARACTER TO REAL VARIABLE
C          I4FCTN     ADAS      CONVERTS FROM CHAR. TO INTEGER  VARIABLE
C          XXWORD     ADAS      PARSES A STRING INTO SEPARATE WORDS
C                               FOR ' ()<>{}' DELIMITERS
C
C AUTHOR:  HP SUMMERS   (REVISION OF BXDATA BY PE BRIDEN)
C          K1/1/57
C          JET EXT. 4941
C
C DATE:    11/06/92
C
C UPDATE:   9/07/93  HPS - USE NEW VERSIONS OF PARSING ROUTINES
C                          B8PARS AND B8PRS1
C UPDATE:  12/07/93  HPS - REVISE TO CONSISTENCY WITH  BXDATA
C                          AT 25/07/93.
C UPDATE:  11/05/95  HPS - ADDED CPRTA TO PARAMETER LIST.ALTERED
C                          'READ()BUFFER' TO BE CONSISTENT WITH IDL-ADAS
C UPDATE:  13/11/95  DHB - INCREASED LENGTH OF CPRTA FROM 4 TO 9 &
C                          STRING FROM 55 TO 75 IN LINE WITH
C                          MODIFICATIONS TO ACCOMODATE J-RESOLVED
C                          PARENT METASTABLES IN THE DATASETS.
C UPDATE:  16/01/96  DHB - INCREASED LENGTH OF CLINE TO 92 & STRG1 TO
C                          56. ALTERED FORMAT NO. 1003 & READING OF
C                          CLINE FORMAT TO ACCOMODATE CHANGES.
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 19-1-96
```

```
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - PUT UNDER SCCS CONTROL
C
C VERSION: 1.2 DATE: 03-07-97
C MODIFIED: RICHARD MARTIN
C - CHANGED I3 TO I4 IN FORMAT STATEMENT 1001
C
C VERSION: 1.3 DATE: 20-11-98
C MODIFIED: DAVID H. BROOKS
C - CHANGED MTIED TO 250.
C
C-----------------------------------------------------------------------
C VERSION: 1.1 DATE: 26-10-99
C MODIFIED: Martin O'Mullane
C - First version.
C
C UPDATE: 1.2 DATE: 17/05/07
C MODIFIED: Allan Whiteford
C - Updated comments as part of subroutine documentation
C                 procedure.
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      CHARACTER           CPLA(NDLEV)
      CHARACTER*9         CPRTA(NDMET)
      CHARACTER*(*)       CSTRGA(NDLEV)
      CHARACTER           TCODE(NDTRN)
      CHARACTER*3         TITLED
      INTEGER             I1A(NDTRN),  I2A(NDTRN),  IA(NDLEV),    IL
      INTEGER             ILA(NDLEV),  IPLA(NDMET,NDLEV)
      INTEGER             ISA(NDLEV),  ITRAN,       IUNIT,        IZ
      INTEGER             IZ0,         IZ1,         MAXLEV,       NDLEV
      INTEGER             NDMET,       NDTRN,       NPL
      INTEGER             NPLA(NDLEV), NV
      LOGICAL             LBSETA(NDMET)
      REAL*8              AVAL(NDTRN), BWNO,        BWNOA(NDMET)
      REAL*8              PRTWTA(NDMET),            SCEF(NVMAX)
      REAL*8              SCOM(NVMAX,NDTRN),        WA(NDLEV)
      REAL*8              XJA(NDLEV),  ZPLA(NDMET,NDLEV)
```

3.43 b8getp: Subroutine b8getp from library adas2xx

```
C
       SUBROUTINE B8GETP(
     &                    IZ0    , IZ1    , DSNEXP, DSNSPF  ,
     &                    NDLEV  , NDMETI , NDTEMI , NDDENI ,
     &                    MAXD   , MAXT   , DENSA  , TEA    ,
     &                    LPDATA , LIOSEL , LRSEL  , LHSEL  ,
     &                    IL     , ITIN   , IDIN   ,
     &                    PCC    , PCIE   , PCIEPR , PV3PR  ,
     &                    PVCRPR , PVECR  , IUNT27 , OPEN27 ,
     &                    PR
     &                  )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: B8GETP *********************
C
C  PURPOSE:  TO FETCH DATA FROM EXPANSION FILE AND CONDENSED BUNDLE-N
C            MATRIX FILE AND COMBINE WITH COLLISIONAL-RADIATIVE
C            DATA FOR IN THE LOW LEVEL POPULATION SOLUTION.
C
C  CALLING PROGRAM: ADAS208
C
C  DATA:
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  IZ0     = NUCLEAR CHARGE
C  INPUT : (I*4)  IZ1     = ION CHARGE+1 (=CHARGE OF PARENT)
C  INPUT : (C*80) DSNEXP  = FULL NAME OF EXPANSION FILE INCLUDING '/UID'
C  INPUT : (C*80) DSNSPF  = FULL NAME OF SPEC. ION FILE READ IN MAIN
C                           PROGRAM INCLUDING '/UID'
C  INPUT : (I*4)  NDLEV   = MAX. NUMBER OF ENERGY LEVELS ALLOWED
C                           IN MAIN PROGRAM
C  INPUT : (I*4)  NDMETI  = MAX. NUMBER OF METASTABLE LEVELS ALLOWED
C                           IN MAIN PROGRAM
C  INPUT : (I*4)  NDTEMI  = MAX. NUMBER OF TEMPERATURES ALLOWED
C                           IN MAIN PROGRAM
C  INPUT : (I*4)  NDDENI  = MAX. NUMBER OF DENSITIES ALLOWED
C                           IN MAIN PROGRAM
C  INPUT : (I*4)  MAXD    = NUMBER OF DENSITIES IN MAIN PROGRAM
C  INPUT : (I*4)  MAXT    = NUMBER OF TEMPERATURES IN MAIN PROGRAM
C  INPUT : (R*8)  DENSA() = SET OF DENSITIES (CM-3) IN MAIN PROGRAM
C  INPUT : (R*8)  TEA()   = SET OF TEMPERATURES (K) IN MAIN PROGRAM
C  INPUT : (L*4)  LPDATA  = .TRUE. - EXPANSION DATA EXISTS AND IS SET
C                           .FALSE.- NO EXPANSION DATA OR NOT SET
C  INPUT : (L*4)  LIOSEL  = .TRUE. - INCLUDE DIRECT IONISATION ON OUTPUT
C                           .FALSE.- DO NOT INCLUDE
C  INPUT : (L*4)  LHSEL   = .TRUE. - INCLUDE ELECTRON RECOM  ON OUTPUT
C                           .FALSE.- DO NOT INCLUDE
C  INPUT : (L*4)  LRSEL   = .TRUE. - INCLUDE CHARGE EXCHANGE ON OUTPUT
C                           .FALSE.- DO NOT INCLUDE
C  INPUT : (I*4)  IL      = INPUT COPASE FILE - NUMBER OF ENERGY LEVELS
C  INPUT : (I*4)  ITIN    = INDEX OF REQUIRED TEMPERATURE IN TEA() SET
C  INPUT : (I*4)  IDIN    = INDEX OF REQUIRED DENSITY IN DENSA() SET
C
C  INPUT : (I*4)  IUNT27  = UNIT FOR PAPER.TEXT OUTPUT
C  INPUT : (L*4)  OPEN27   = .TRUE. - PAPER.TEXT HAS BEEN OPENED
C     .FALSE.- PAPER.TEXT HAS NOT BEEN OPENED
C
C  OUTPUT: (R*8)  PCC(,)  = PROJETED COLL. RAD. LOW LEVEL MATRIX
```

```
C                                1ST DIM: ENERGY LEVEL INDEX
C                                2ND DIM: ENERGY LEVEL INDEX
C  OUTPUT: (R*8)  PCIE()  = PROJECTED COLL. RAD. ION. COEFFT. VECTOR
C                                1ST DIM: ENERGY LEVEL INDEX
C  OUTPUT: (R*8)  PCIEPR(,)=PROJECTED PARENT RESOLVED COLL. RAD. ION
C                           MATRIX
C                                1ST DIM: ENERGY LEVEL INDEX
C                                2ND DIM: PARENT INDEX
C  OUTPUT: (R*8)  PV3PR(,)= DIRECT PARENT RESOLVED THREE
C                           BODY RECOMB. COEFFT MATRIX
C                                1ST DIM: ENERGY LEVEL INDEX
C                                2ND DIM: PARENT INDEX
C                                UNITS : CM3S-1
C  OUTPUT: (R*8)  PVECR(,)= PROJECTED PARENT RESOLVED COLL. RAD.
C                           RECOMB. COEFFT MATRIX ( RR + DR + 3B )
C                                1ST DIM: ENERGY LEVEL INDEX
C                                2ND DIM: PARENT INDEX
C                                UNITS : CM3S-1
C  OUTPUT: (R*8)  PR( ,,) = RECOM/BREMS. COEFFT (ERG S-1)
C                                1ST DIM: PARENT INDEX
C                                2ND DIM: TEMPERATURE INDEX
C                                3RD DIM: DENSITY INDEX
C
C
C          (C*80) DSNCPM  = FULL NAME OF COND.MAT. FILE INCLUDING '/UID'
C                           EXPANDED IF NECESSARY FROM SYMBOLIC FILENAME
C                           IN NAMELIST IN EXPANSION FILE
C          (C*80) DSNREF  = FULL NAME OF SPEC. ION FILE INCLUDING '/UID'
C                           EXPANDED IF NECESSARY FROM SYMBOLIC FILENAME
C                           IN NAMELIST IN EXPANSION FILE
C          (C*80) DSHORT  = TEMPORARY STRING
C          (C*11) PTSYMA()= PARENT SYMMETRY (2SP+1 LP) AS CHARACTERS
C                             1ST DIMENSION: PARENT INDEX
C          (I*4)  NPTSPA()= PARENT SPIN (2SP+1)
C                             1ST DIMENSION: PARENT INDEX
C          (I*4)  NSPSYS()= NO. OF SPIN SYSTEMS ASSOCIATED WITH PARENT
C                             1ST DIMENSION: PARENT INDEX
C          (I*4)  NCUTP() = N-SHELL CUT-OFF ASSOCIATED WITH AUGER
C                           PROCESSES FOR THE PARENT
C                             1ST DIMENSION: PARENT INDEX
C          (R*8)  DEPA()  = BINDING ENERGY (RYD) OF LOWEST AUGER
C                           N-SHELL FOR THE PARENT
C                             1ST DIMENSION: PARENT INDEX
C          (I*4)  NSHEL   = NUMBER OF N-SHELLS INVOLVED IN EXPANSION
C          (I*4)  NSHELA()= N-SHELLS INVOLVED IN THE EXPANSION
C                             1ST DIMENSION: SHELL INDEX (<= NSHEL)
C          (I*4)  NSPIN   = NUMBER OF SPIN SYSTEMS FOR CURRENT PARENT
C          (I*4)  NSPNA(,)= SPIN OF SYSTEM (2S+1)
C                             1ST DIMENSION: SPIN SYSTEM INDEX
C                             2ND DIMENSION: PARENT INDEX
C          (I*4)  NLWSTA(,)= LOWEST N-SHELL INCLUDED FOR THE SPIN SYSTEM
C                             1ST DIMENSION: SPIN SYSTEM INDEX
C                             2ND DIMENSION: PARENT INDEX
C          (R*8)  PLWSTA(,)= PHASE SPACE OCCUPANCY FACTOR FOR LOWEST
C                            N-SHELL FOR SPIN SYSTEM
C                             1ST DIMENSION: SPIN SYSTEM INDEX
C                             2ND DIMENSION: PARENT INDEX
C          (R*8)  FLWSTA(,)= FRACTIONAL PARENTAGE (EQUIV. ELECTRONS) FOR
C                            FOR IONISATION FROM LOWEST LEVEL OF
C                            SPIN SYSTEM
C                             1ST DIMENSION: SPIN SYSTEM INDEX
```

```
C                              2ND DIMENSION: PARENT INDEX
C          (R*8)  FRACPRT  = TEMP. STORE OF FRACTIONAL PARENTAGE
C          (I*4)  INDA()   = LEVEL INDEX WITH RESPECT TO SPEC. ION FILE
C                              1ST DIMENSION: COUNTER OVER EXP. RECORDS
C          (C*11) LVSYMA()= LEVEL SYMMETRY AND ADDITIONAL INFO.ON CONFIG
C                              1ST DIMENSION: COUNTER OVER EXP. RECORDS
C          (I*4)  LSZDA() = SZD FILE SELECTOR FOR RECORD (IF REQUIRED)
C                              1ST DIMENSION: COUNTER OVER EXP. RECORDS
C          (I*4)  LSPA()   = SPIN SYSTEM (2S+1) FOR RECORD
C                              1ST DIMENSION: COUNTER OVER EXP. RECORDS
C          (I*4)  LSHA()   = ACTIVE N SHELL FOR RECORD
C                              1ST DIMENSION: COUNTER OVER EXP. RECORDS
C          (I*4)  LPTA()   = PARENT INDEX  FOR RECORD
C                              1ST DIMENSION: COUNTER OVER EXP. RECORDS
C          (R*8)  WGHTA(,)= WEIGHTING FOR EXPANSION FOR RECORD
C                              1ST DIMENSION: COUNTER OVER EXP RECORDS
C                              2ND DIMENSION: NSHELL INDEX
C          (I*4)  NMET     = NUMBER OF '*' LEVELS COUNTED
C                            (NB. USE ONLY ONCE FOR A GIVEN LEVEL
C                                 EVEN THOUGH ANOTHER RECORD FOR THE
C                                 LEVEL MAY EXIST)
C          (I*4)  IMETR() = LEVEL INDEX OF METASTABLES '*'ED
C                              1ST DIMENSION: METASTABLE COUNTER(<=NMET)
C          (C*250)LSTRNG   = COND. BUNDLE-N. MATRIX (CBNM) FILE RECORD
C          (C*2)  SEQM     = SEQUENCE IDENTIFIER GIVEN ON CBNM FILE
C          (I*4)  NUCGM    = NUCLEAR CHARGE GIVEN ON CBNM FILE
C          (I*4)  NPRTM    = NO. OF PARENTS GIVEN ON CBNM FILE
C          (I*4)  MAXDM    = NO. OF DENSITIES GIVEN ON CBNM FILE
C          (I*4)  MAXTM    = NO. OF TEMPERATURES GIVEN ON CBNM FILE
C          (I*4)  IPRT     = PARENT INDEX
C          (I*4)  IPRTM    = PARENT INDEX IN CBNM FILE
C          (C*4)  TRMPM    = PARENT TERM SPECIFICATION AS (2SP+1LP)
C          (I*4)  SPNPM    = PARENT SPIN (2SP+1)
C          (I*4)  ISYSM    = SPIN SYSTEM INDEX IN CBNM FILE
C          (I*4)  SSYSM(,)= SPIN SYSTEM IN CBNM FILE
C                              1ST DIM.: PARENT INDEX       (<=NDMET)
C                              2ND DIM.: SPIN SYSTEM INDEX  (<=2)
C          (I*4)  NSYSM() = NO OF SPIN SYSTEM IN CBNM FILE FOR PARENT
C                              1ST DIM.: PARENT INDEX       (<=NDMET)
C          (I*4)  NSHLM(,)= NO. OF N-SHELLS IN CBNM FILE
C                              1ST. DIM.: PARENT INDEX      (<=NDMET)
C                              2ND. DIM.: SPIN SYSTEM INDEX (<=2)
C          (R*8)  DENSM() = ELECTRON DENSITIES (CM-3) ON CBNM FILE
C                              1ST DIMENSION: DENSITY INDEX (<=NDMAX)
C          (R*8)  TEM()    = ELECTRON TEMPS.  (K) ON CBNM FILE
C                              1ST DIMENSION: TEMP. INDEX (<=NTMAX)
C          (R*8)  PCRMAT(,,,,,)=PROJECTED COLLISIONAL-RADIATIVE MATRIX
C                                IN P-REPRESENTATION WITHOUT ELIMINATIONS
C                                1ST DIM.: TEMPERATURE INDEX
C                                2ND DIM.: DENSITY INDEX
C                                3RD DIM.: ROW INDEX
C                                4TH DIM.: COLUMN INDEX
C                                5TH DIM.: PARENT INDEX
C                                6TH DIM.: SPIN SYSTEM INDEX
C          (R*8)  PIOMAT(,,,,,)=PROJECTED COLLISIONAL-RADIATIVE IONIS.
C                                MATRIX TO RESOLVED + METASTABLES
C                                IN P-REPRESENTATION WITHOUT ELIMINATIONS
C                                1ST DIM.: TEMPERATURE INDEX
C                                2ND DIM.: DENSITY INDEX
C                                3RD DIM.: ROW INDEX
C                                4TH DIM.: COLUMN INDEX (+ METASTABLES)
```

```
C                               5TH DIM.: PARENT INDEX
C                               6TH DIM.: SPIN SYSTEM INDEX
C          (R*8)  PQPIND(,,,,,)=PROJECTED INDIRECT PARENT CQ COEFFICIENT
C                               MATRIX FROM SPECIFIC PARENT, SPIN TO
C                               FINAL PARENT IN PN REPRESENTATION
C                               1ST DIM.: TEMPERATURE INDEX
C                               2ND DIM.: DENSITY INDEX
C                               3TH DIM.: FINAL   PARENT INDEX
C                               4TH DIM.: INITIAL PARENT INDEX
C                               5TH DIM.: SPIN SYSTEM INDEX
C          (R*8)  PVCRPR(,,)   =PROJECTED INDIRECT PARENT CQ COEFFICIENT
C                               MATRIX FROM SPECIFIC PARENT TO
C                               FINAL PARENT IN PN REPRESENTATION
C                               SUMMED OVER SPIN SYSTEMS
C                               1ST DIM.: FINAL   PARENT INDEX
C                               2ND DIM.: INITIAL PARENT INDEX
C          (R*8)  PCRRHS(,,,,)= PROJECTED COLLISIONAL-RADIATIVE RECOM.
C                               RHS. FROM A SPECIFIED PARENT AND IN
C                               A SPECIFIED SPIN SYSTEM
C                               IN P-REPRESENTATION WITHOUT ELIMINATIONS
C                               1ST DIM.: TEMPERATURE INDEX
C                               2ND DIM.: DENSITY INDEX
C                               3RD DIM.: ROW INDEX
C                               5TH DIM.: PARENT INDEX
C                               6TH DIM.: SPIN SYSTEM INDEX
C          (R*8)  PRB(,,,)     = RECOM/BREMS. COEFFT (
C                               1ST DIM: TEMPERATURE INDEX
C                               2ND DIM: DENSITY INDEX
C                               3RD DIM: PARENT INDEX
C                               4TH DIM: SPIN SYSTEM INDEX
C          (R*8)  DCRMAT(,,,) = DIRECT COLLISIONAL-RADIATIVE MATRIX
C                               IN P-REPRESENTATION FOR LOW N-SHELLS
C                               1ST DIM.: TEMPERATURE INDEX
C                               2ND DIM.: DENSITY INDEX
C                               3RD DIM.: ROW INDEX
C                               4TH DIM.: COLUMN INDEX
C          (R*8)  DIOMAT(,,,) = DIRECT COLLISIONAL-RADIATIVE IONIS.
C                               MATRIX TO RESOLVED + METASTABLES
C                               IN P-REPRESENTATION FOR LOW N-SHELLS
C                               1ST DIM.: TEMPERATURE INDEX
C                               2ND DIM.: DENSITY INDEX
C                               3RD DIM.: ROW INDEX
C                               4TH DIM.: COLUMN INDEX (+ METASTABLES)
C          (R*8)  DTREC(,)    =  DIRECT THREE-BODY RECOMBINATION COEFFTS.
C                               FROM A SPECIFIED PARENT AND IN A
C                               SPECIFIED SPIN SYSTEM
C                               1ST DIM.: TEMPERATURE INDEX
C                               2ND DIM.: ROW INDEX
C          (R*8)  DDREC(,)    =  DIRECT DIELECTR.  RECOMBINATION COEFFTS.
C                               FROM A SPECIFIED PARENT AND IN A
C                               SPECIFIED SPIN SYSTEM
C                               1ST DIM.: TEMPERATURE INDEX
C                               2ND DIM.: ROW INDEX
C          (R*8)  DRREC(,)    =  DIRECT RADIATIVE  RECOMBINATION COEFFTS.
C                               FROM A SPECIFIED PARENT AND IN A
C                               SPECIFIED SPIN SYSTEM
C                               1ST DIM.: TEMPERATURE INDEX
C                               2ND DIM.: ROW INDEX
C          (R*8)  DXREC(,)    =  DIRECT CH. EXCH.  RECOMBINATION COEFFTS.
C                               FROM A SPECIFIED PARENT AND IN A
C                               SPECIFIED SPIN SYSTEM DUE TO H(1S)
```

```
C                                1ST DIM.: TEMPERATURE INDEX
C                                2ND DIM.: ROW INDEX
C          (I*4)  NM()       =  LOW N-SHELLS FOR PARENT SPIN SYSTEM
C                                COMBINATION
C                                1ST. DIM.: N-SHELL INDEX
C          (I*4)  NSUP(,)    =  HIGHEST N-SHELL REQUIRED FOR EXPANSION
C                                FOR THE PARENT AND SPIN SYSTEM
C                                1ST. DIM.: PARENT INDEX
C                                2ND. DIM.: SPIN SYSTEM INDEX
C          (I*4)  ISPIN      =  GENERAL INDEX
C          (I*4)  IPT        =  GENERAL INDEX
C          (I*4)  JPT        =  GENERAL INDEX
C          (I*4)  I          =  GENERAL INDEX
C          (I*4)  J          =  GENERAL INDEX
C          (I*4)  II         =  GENERAL INDEX
C          (I*4)  JJ         =  GENERAL INDEX
C          (I*4)  IR         =  GENERAL INDEX
C          (I*4)  IC         =  GENERAL INDEX
C          (I*4)  IS         =  GENERAL INDEX
C          (I*4)  KI         =  GENERAL INDEX
C          (I*4)  KJ         =  GENERAL INDEX
C          (I*4)  IN         =  DENSITY INDEX
C          (I*4)  IT         =  TEMPERATURE INDEX
C          (I*4)  NUP        =  UPPER N-SHELL FOR CURRENT EXPANSION
C          (I*4)  IMAX       =  NO. OF SHELLS REQUIRED IN EXPANSION
C          (L*4)  LSOLVE     = .TRUE. -INVERSION WITH SOLN. OF EQUATIONS
C                            = .FALSE. -INVERSION ONLY
C          (R*8)  AMAT(,)    = TEMPORARY ARRAY FOR INVERTING
C          (R*8)  BRHS()     = TEMPORARY R.H.S FOR EQUATION SOLUTION
C          (R*8)  DINTX      = + OR - DEPENDING ON INTERCHANGES IN
C                              INVERSION SUBROUTINE XXMINV
C          (R*8)  PCRTMP(,)  = TEMPORARY PROJECTED COLL. RAD. MATRIX
C                              TO BE CONDENSED TO PCRMAT
C          (R*8)  DCRTMP(,)  = TEMPORARY DIRECT COLL. RAD. MATRIX
C                              TO BE CONDENSED TO DCRMAT
C          (R*8)  PIOTMP(,)  = TEMPORARY PROJECTED IONIS. MATRIX
C                              TO BE CONDENSED TO PIOMAT
C          (R*8)  PRHTMP(,)  = TEMPORARY PROJECTED R.H.S. VECTOR
C                              TO BE CONDENSED TO PCRRHS
C          (R*8)  PQPTMP()   = TEMPORARY INDIRECT PARENT QC COEFFICIENT
C                              TO BE CONDENSED TO PQPIND
C                                 1ST INDEX  - FINAL PARENT
C          (R*8)  SUM        = GENERAL USE FOR SUMMING
C          (R*8)  Z0         = NUCLEAR CHARGE
C          (R*8)  Z1         = ION CHARGE+1 (=CHARGE OF PARENT)
C          (R*8)  SSYSWT     = FRACTIONAL WEIGHTING OF SPIN SYSTEM
C                              FOR PARTICULAR PARENT TO BE USED IF
C                              RECOMBINATION COEFFICIENTS ARE GIVEN IN
C                              THE MULTIPLIED UP FORM.
C          (L*4)  LTRNG()    = .TRUE. - OUTPUT VALUE WAS EXTRAPOLATED
C                                       FOR TEMPERATURE
C                              .FALSE. - OUTPUT VALUE NOT EXTRAPOLATED
C          (L*4)  LDRNG()    = .TRUE. - OUTPUT VALUE WAS EXTRAPOLATED
C                                       FOR DENSITY
C                              .FALSE. - OUTPUT VALUE NOT EXTRAPOLATED
C          (I*4)  IUP        = NUP-NM(1)+1
C          (R*8)  V          = TEMPORARY REAL NUMBER
C          (R*8)  ARRIN(,)   = TEMPORARY ARRAY FOR INPUT TO SPLINING
C          (R*8)  ARROUT(,)  = TEMPORARY ARRAY FOR IOUTPUT FROM SPLINING
C          (R*8)  TEMIN      = MINIMUM TEMPERATURE BELOW WHICH COEFFT.
C                              SHOULD BE SET TO ZERO
```

```
C          (R*8)  DEMIN      = MINIMUM DENSITY  BELOW WHICH COEFFT.
C                               SHOULD BE SET TO ZERO
C          (R*8)  DETMP      = TEMPORARY VALUE OF DEMIN
C          (R*8)  TETMP      = TEMPORARY VALUE OF TEMIN
C                               SHOULD BE SET TO ZERO
C          (I*4)  IUPA(,)    = DIMENSION OF FINAL CONDENSED N-SHELL
C                               MATRIX
C                               1ST DIM: PARENT INDEX
C                               2ND DIM: SPIN SYSTEM INDEX
C C        (I*4)  IPOINTA()  = POINTER TO INDEX OF N-SHELL IN NSHEL
C C                             LIST
C C                             1ST DIM: N=PRINCIPAL QUANTUM NUMBER
C
C          (I*4)  IEDMAT   = 0 PCRL ADDED ONTO PCRMAT
C                            1 PCRL NOT ADDED ON
C          (I*4)  IECION   = 0 PCION  ADDED ONTO TO PCRMAT
C                              PCIONRI ADDED ONTO PCIONRP
C                            1 PCION NOT ADDED ON
C                              PCIONRI NOT ADDED ON
C          (I*4)  IETREC   = 0 PTREC ADDED ONTO PCRRHS
C                            1 PTREC NOT ADDED ON
C          (I*4)  IEDREC   = 0 PDREC ADDED ONTO PCRRHS
C                            1 PDREC NOT ADDED ON
C          (I*4)  IERREC   = 0 PRREC ADDED ONTO PCRRHS
C                            1 PRREC NOT ADDED ON
C          (I*4)  IEXREC   = 0 PXREC ADDED ONTO PCRRHS
C                            1 PXREC NOT ADDED ON
C          (I*4)  IERSYS   = 0 RECOMBINATION RATES MULTIPLIED
C                                  BY SPIN SYSTEM WEIGHT
C                            1 RECOMBINATION RATES NOT MULTIPLIED
C                                  BY SSYSWT
C
C ROUTINES: NONE
C
C STREAM HANDLING :
C           7     OUTPUT (PAPER.TEXT)
C          14     EXPANSION FILE
C          15     CONDENSED MATRIX MASTER FILE
C
C AUTHOR:  HP SUMMERS
C          K1/1/57
C          JET EXT. 4941
C
C DATE:    18/08/92
C
C-----------------------------------------------------------------------
C
C UPDATE:  WJ DICKSON
C          K1/1/26
C
C   DATE:  JANUARY 1993
C
C          NUMEROUS ADJUSTMENTS AND UPDATES
C
C-----------------------------------------------------------------------
C
C UPDATE:  WJ DICKSON
C          K1/1/26
C
C   DATE:  12TH AUGUST 1993
C
```

```
C          INCLUSION OF VARIABLES IEFPRS AND IEFPRE AND CORRESPONDING
C          ADJUSTMENTS TO DIO , PCR AND PIO MATRICES. FRACTIONAL
C          PARENTAGE COEFFICIENTS AS GIVEN BY EXPANSION FILE
C          ( VARIABLE  FLWSTA )
C
C          (I*4)  IEFPRS   = 0 GROUND STATE IONISATION RATE COEFFICIENTS
C                              HAVE BEEN MULTIPLYIED BY FRACTIONAL
C                              PARENTAGE COEFFICIENT IN MAINBNS
C                            1 GROUND STATE IONISATION RATE COEFFICIENTS
C                              HAVE NOT BEEN MULTIPLYIED BY FRACTIONAL
C                              PARENTAGE COEFFICIENT IN MAINBNS
C
C          (I*4)  IEFPRE   = 0 ELEMENTS OF MAIN C-R MATRIX ARISING
C                              FROM GROUND STATE
C                              HAVE BEEN MULTIPLYIED BY FRACTIONAL
C                              PARENTAGE COEFFICIENT IN MAINBNS
C                            1 ELEMENTS OF MAIN C-R MATRIX ARISING
C                              FROM GROUND STATE
C                              HAVE NOT BEEN MULTIPLYIED BY FRACTIONAL
C                              PARENTAGE COEFFICIENT IN MAINBNS
C
C UNIX-IDL PORT:
C
C DATE: UNKNOWN
C
C AUTHOR: DAVID H BROOKS, UNIVERSITY OF STRATHCLYDE
C
C UPDATE:  29/03/96  HPS - INCREASE PARAMETER SETTINGS NDTEM: 20->35
C                                                      NDDEN: 20->24
C
C UPDATE:  18/04/96  HPS - ALTER FORMAT 2008 FOR READING TEMPS. AND
C                          DENS. FROM CBNM FILE FOR CONSISTENCY WITH
C                          NEW PRODUCTION VERSION OF ADAS204
C UPDATE:  18/04/96  HPS - ALTER B8SPLNX TO B8SPLN IN 2ND AND 3RD
C                          CALLS IN THE SUBROUTINE
C UPDATE:  03/05/96  DHB - ALTERED IBM SPECIFIC STATEMENTS. INCREASED
C                          SIZE OF DSNINC & DSNSPF TO 80.
C UPDATE:  09/03/98  HPS - ADDED PR TO PARAMETER LIST. PREPARED FROM PRB
C                          FROM PROJECTION MATRIX FILE BY INTERPOLATION.
C                          CORRECTED PB TO INCLUDE SUM OVER SPIN SYSTEMS
C**********************************************************************
C PUT UNDER SCCS CONTROL:
C
C DATE:    10-05-96
C
C VERSION: 1.1 DATE: 10-05-96
C MODIFIED: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C     - FIRST PUT UNDER SCCS
C
C VERSION: 1.2
C MODIFIED: WILLIAM OSBORN DATE: 13-05-96
C     - ADDED IUNT27 AND OPEN27 TO ALLOW PAPER.TEXT OUTPUT
C
C VERSION: 1.3
C MODIFIED: WILLIAM OSBORN + HPS DATE: 28-05-96
C     - ADDED CALL TO XXFLNM TO EXPAND FILENAMES
C
C VERSION: 1.4
C MODIFIED: TIM HAMMOND                 DATE: 02-08-96
C           - CHANGED NAME OF VARIABLE DINT TO DINTX AS DINT IS THE
C             NAME OF AN INTRINSIC FUNCTION ON HP WORKSTATIONS
```

```
C
C VERSION: 1.5
C MODIFIED: RICHARD MARTIN                DATE: 02-03-98
C           - CHANGED IUNT7 TO IUNT27 AND OPEN7 TO OPEN27.
C
C VERSION: 1.6
C MODIFIED: HUGH SUMMERS                  DATE: 09-03-98
C - ADDED PR TO PARAMETER LIST. PREPARED FROM PRB
C             FROM PROJECTION MATRIX FILE BY INTERPOLATION.
C             CORRECTED PB TO INCLUDE SUM OVER SPIN SYSTEMS
C
C VERSION:   1.7                                DATE: 2/09/99
C MODIFIED:  Martin O'Mullane
C            - Format error in 2020. Skip 70X not 80X.
C
C VERSION:   1.8                                DATE: 26/10/99
C MODIFIED:  Martin O'Mullane
C            - Added call to b8splt to correct for odd behaviour
C              at low ne high Te recombination data.
C
C VERSION:   1.9                                DATE: 08/12/99
C MODIFIED:  Martin O'Mullane
C            - Added check for expansion data which did not have any
C              weighting factors. Only proceed if LPTA is non zero.
C              It is the do 56 loop.
C            - The assumption that variables are saved between calls
C              should not be made (Linux again). Anyway it's bad
C              practice. Data required when called with LPDATA true
C              is now saved.
C
C VERSION:   1.10                                DATE: 13/02/2006
C MODIFIED:  Martin O'Mullane
C            - Increase number of levels to 150.
C            - Write status of projection to screen.
C            - Index error in FLWSTA corrected.
C
C-----------------------------------------------------------------------
      CHARACTER*80        DSNEXP,      DSNSPF
      INTEGER             IDIN,        IL,          ITIN,        IUNT27
      INTEGER             IZ0,         IZ1,         MAXD,        MAXT
      INTEGER             NDDENI,      NDLEV,       NDMETI,      NDTEMI
      LOGICAL             LHSEL,       LIOSEL,      LPDATA,      LRSEL
      LOGICAL             OPEN27
      REAL*8              DENSA(NDDENI),            PCC(NDLEV,NDLEV)
      REAL*8              PCIE(NDLEV), PCIEPR(NDLEV,NDMETI)
      REAL*8              PR(NDMETI,NDTEMI,NDDENI), PV3PR(NDLEV,NDMETI)
      REAL*8              PVCRPR(NDMETI,NDMETI),    PVECR(NDLEV,NDMETI)
      REAL*8              TEA(NDTEMI)
```

3.44 b8loss: Subroutine b8loss from library adas2xx

```
C
      SUBROUTINE B8LOSS( NDTRN  , NDLEV  , NDMET ,
     &                   ICNTE  , NMET   , IMETR , ISTRN  ,
     &                   XJA    , ER     , AVAL  ,
     &                   IE1A   , IE2A   ,
     &                   SLOSS  , SWVLN  , TLOSS
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: B8LOSS *********************
C
C  PURPOSE: TO CALCULATE THE DIRECT LINE POWER LOSS FOR EACH LEVEL AND
C           IDENTIFY THE STRONGEST SPECIFIC LINE POWER TRANSITIONS TO
C           EACH METASTABLE LEVEL.
C
C           (MODIFICATION OF B6LOSS)
C
C  CALLING PROGRAM:  ADAS208
C
C  SUBROUTINE:
C
C  INPUT :  (I*4)  NDTRN   = MAXIMUM NUMBER OF TRANSITIONS ALLOWED
C  INPUT :  (I*4)  NDLEV   = MAXIMUM NUMBER OF ENERGY LEVELS ALLOWED
C  INPUT :  (I*4)  NDMET   = MAXIMUM NUMBER OF METASTABLES ALLOWED
C
C  INPUT :  (I*4)  ICNTE   = NUMBER OF ELECTRON IMPACT TRANSITIONS
C  INPUT :  (I*4)  NMET    = NUMBER OF METASTABLES
C  INPUT :  (I*4)  IMETR() = METASTABLE INDICES IN LEVEL LIST
C                            DIMENSION: METASTABLE COUNT INDEX
C  OUTPUT:  (I*4)  ISTRN() = SPECIFIC LINE POWER: SELECTED ELECTRON
C                            IMPACT TRANSITION INDEX. (FOR USE WITH
C                            'IE1A()' , 'IE2A()' AND 'AA()' ARRAYS)
C                            WHICH GIVES LARGEST POWER TO METASTABLE
C                            DIMENSION: METASTABLE COUNT INDEX
C
C
C  INPUT :  (R*8)  XJA()   = QUANTUM NUMBER (J-VALUE) FOR GIVEN LEVEL.
C                            NOTE: (2*XJA)+1 = STATISTICAL WEIGHT
C  INPUT :  (R*8)  ER()    = ENERGY RELATIVE TO LEVEL 1 (RYDBERGS)
C                            DIMENSION: ENERGY LEVEL.
C  INPUT :  (R*8)  AVAL()  = ELECTRON IMPACT TRANSITION: A-VALUE (SEC-1)
C                            DIMENSION: ENERGY LEVEL.
C
C  INPUT :  (I*4)  IE1A()  = ELECTRON IMPACT TRANSITION:
C                            LOWER ENERGY LEVEL INDEX
C  INPUT :  (I*4)  IE2A()  = ELECTRON IMPACT TRANSITION:
C                            UPPER ENERGY LEVEL INDEX
C
C  OUTPUT:  (R*8)  SLOSS()  = DIRECT LINE POWER LOSS FOR SPECIFIC LINE
C                             POWER TRANSITION GIVEN BY 'ISTRN' FOR EACH
C                             METASTABLE (UNITS: ERGS SEC-1)
C                             DIMENSION: METASTABLE COUNT INDEX
C  OUTPUT:  (R*8)  SWVLN()  = WAVELENGTH (ANGSTROM) FOR SPECIFIC LINE
C                             POWER TRANSITION GIVEN BY 'ISTRN' FOR EACH
C                             METASTABLE (UNITS: ERGS SEC-1)
C                             DIMENSION: METASTABLE COUNT INDEX
C  OUTPUT:  (R*8)  TLOSS()  = DIRECT LINE POWER LOSS FOR EACH LEVEL.
C                             (UNITS: ERGS SEC-1)
C                             DIMENSION: LEVEL INDEX
```

```
C
C           (R*8)  R2LOSS  = PARAMETER = EQUATION CONSTANT = 2.17958D-11
C                            (CONVERTS RYDBERGS/SEC TO ERGS/SEC)
C           (R*8)  WCVRN   = PARAMETER = EQUATION CONSTANT = 911.268
C                            (CONVERTS RYD. TRANS ENERGY TO WAVELENGTH
C                             IN ANGSTROM)
C           (R*8)  SCURR   = CURRENT INDIVIDUAL LINE POWER
C
C           (I*4)  LLOWER  = SELECTED ELECTRON IMPACT TRANSITION:
C                            LOWER ENERGY LEVEL INDEX
C           (I*4)  LUPPER  = SELECTED ELECTRON IMPACT TRANSITION:
C                            UPPER ENERGY LEVEL INDEX
C           (I*4)  IC      = TRANSITION ARRAY INDEX
C           (I*4)  IM      = METASTABLE COUNT INDEX
C
C
C ROUTINES:  NONE
C
C NOTES:
C            EQUATIONS USED -
C
C            FOR EACH TRANSITION - DIRECT LINE POWER LOSS IS GIVEN BY:
C
C            LOSS  = 'R2LOSS' x AVALUE x (ENERGY DIFFERENCE)
C
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/37
C          JET EXT. 5023
C
C DATE:    09/10/90
C
C UPDATE:  29/07/92 - CORRECT ERROR - ZERO TLOSS OVER NDLEV INSTEAD OF
C                                     ICNTE.
C UPDATE:  23/05/96 - CONVERTED B6LOSS TO B8LOSS, CHANGED ISTRN TO
C                     OUTPUT INDEX OF STRONGEST RADIATING TRANSITION TO
C                     EACH METASTABLE.
C
C*********************************************************************
C PUT UNDER S.C.C.S CONTROL:
C
C VERSION: 1.1 DATE: 15/07/96
C MODIFIED: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C     - FIRST PUT UNDER S.C.C.S
C
C VERSION: 1.2 DATE: 20/07/07
C MODIFIED: Allan Whiteford
C     - Small modification to comments to allow for automatic
C             documentation preparation.
C
C---------------------------------------------------------------------
C
C---------------------------------------------------------------------
      INTEGER             ICNTE,       IE1A(NDTRN), IE2A(NDTRN)
      INTEGER             IMETR(NDMET),             ISTRN(NDMET)
      INTEGER             NDLEV,       NDMET,       NDTRN,       NMET
      REAL*8              AVAL(NDTRN), ER(NDLEV),   SLOSS(NDMET)
      REAL*8              SWVLN(NDMET),             TLOSS(NDLEV)
      REAL*8              XJA(NDLEV)
```

3.45 b8mcca: Subroutine b8mcca from library adas2xx

```
C
      SUBROUTINE B8MCCA( NDLEV  , IL     ,
     &                   LPSEL  , LISEL  , LPDATA,
     &                   DENE   , DENP   ,
     &                   CRA    , PCC    ,
     &                   CRCE   , CRCP   , CIE   ,
     &                   CC
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: B8MCCA *********************
C
C  PURPOSE: TO  CONSTRUCT WHOLE RATE MATRIX 'CC' FOR TRANSITIONS BETWEEN
C           ALL ENERGY LEVELS AT A FIXED TEMPERATURE  AND  GIVEN DENSITY
C           'DENE/DENP'.
C
C          UPDATED VERSION OF BXMCCA, TO INCORPORATE INDIRECT COUPLINGS
C
C  CALLING PROGRAM:  ADAS205/ADAS206
C
C  SUBROUTINE:
C
C  INPUT :  (I*4)  NDLEV   = MAXIMUM NUMBER OF ENERGY LEVELS ALLOWED
C  INPUT :  (I*4)  IL      = NUMBER OF ENERGY LEVELS
C
C  INPUT :  (L*4)  LPSEL   = .TRUE.  => INCLUDE PROTON COLLISIONS
C                            .FALSE. => DO NOT INCLUDE PROTON COLLISIONS
C  INPUT :  (L*4)  LISEL   = .TRUE.  => INCLUDE IONISATION RATES
C                            .FALSE. => DO NOT INCLUDE IONISATION RATES
C  INPUT :  (L*4)  LPDATA  = .TRUE.  => INCLUDE INDIRECT COUPLINGS
C                            .FALSE. => DO NOT INCLUDE INDIRECT COUPLING
C
C  INPUT :  (R*8)  DENE    = ELECTRON DENSITY (UNITS: CM-3)
C  INPUT :  (R*8)  DENP    = PROTON DENSITY (UNITS: CM-3)
C
C  INPUT :  (R*8)  CRA(,)  = A-VALUE (sec-1)  MATRIX  COVERING   ALL
C                            TRANSITIONS.
C                            1st DIMENSION: ENERGY LEVEL INDEX
C                            2nd DIMENSION: ENERGY LEVEL INDEX
C                           (NOTE: DIAGONAL   ELEMENTS  REPRESENT   THE
C                                  NEGATIVE  SUM  OF  THEIR  RESPECTIVE
C                                  COLUMNS.)
C
C  INPUT :  (R*8)  PCC(,)  = INDIRECT RATE MATRIX COVERING ALL
C                            TRANSITIONS (UNITS: SEC-1)
C                            VALUES FOR GIVEN TEMPERATURE AND DENSITY.
C                            1st DIMENSION: ENERGY LEVEL INDEX
C                            2nd DIMENSION: ENERGY LEVEL INDEX
C
C  INPUT :  (R*8)  CRCE(,) = ELECTRON IMPACT TRANSITIONS:
C                            EXCIT'N/DE-EXCIT'N RATE COEFFT MATRIX
C                            COVERING ALL TRANSITIONS (cm**3/s).
C                            VALUES FOR GIVEN TEMPERATURE.
C                            1st DIMENSION: ENERGY LEVEL INDEX
C                            2nd DIMENSION: ENERGY LEVEL INDEX
C                           (NOTE: DIAGONAL   ELEMENTS  REPRESENT   THE
C                                  NEGATIVE  SUM  OF  THEIR  RESPECTIVE
C                                  COLUMNS.)
C  INPUT :  (R*8)  CRCP(,) = PROTON IMPACT TRANSITIONS:
```

```
C                              EXCIT'N/DE-EXCIT'N RATE COEFFT MATRIX
C                              COVERING ALL TRANSITIONS (cm**3/s).
C                              VALUES FOR GIVEN TEMPERATURE.
C                              1st DIMENSION: ENERGY LEVEL INDEX
C                              2nd DIMENSION: ENERGY LEVEL INDEX
C                             (NOTE: DIAGONAL   ELEMENTS  REPRESENT   THE
C                                    NEGATIVE  SUM  OF  THEIR  RESPECTIVE
C                                    COLUMNS.)
C  INPUT :  (R*8)  CIE()    = IONISATION RATE COEFFICIENT VECTOR FOR
C                              FIXED TEMPERATURE.
C                              DIMENSION: ENERGY LEVEL INDEX
C
C  OUTPUT:  (R*8)  CC(,)     = RATE MATRIX COVERING ALL TRANSITIONS
C                              (UNITS: SEC-1)
C                              VALUES FOR GIVEN TEMPERATURE AND DENSITY.
C                              1st DIMENSION: ENERGY LEVEL INDEX
C                              2nd DIMENSION: ENERGY LEVEL INDEX
C
C           (I*4)  IS1      = ENERGY LEVEL ARRAY INDEX
C           (I*4)  IS2      = ENERGY LEVEL ARRAY INDEX
C
C
C ROUTINES: NONE
C
C
C AUTHOR:  WILLIAM J. DICKSON (MOSTLY COPIED FROM BXMCCA)
C          K1/1/36
C          JET EXT. 5057
C
C DATE:    06/01/92
C
C UNIX-IDL PORT:
C
C DATE: UNKNOWN
C
C AUTHOR: DAVID H BROOKS, UNIVERSITY OF STRATHCLYDE
C
C**********************************************************************
C PUT UNDER SCCS CONTROL:
C
C VERSION: 1.1 DATE: 10/05/96
C MODIFIED: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C     - FIRST PUT UNDER SCCS
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IL,          NDLEV
      LOGICAL             LISEL,       LPDATA,      LPSEL
      REAL*8              CC(NDLEV,NDLEV),          CIE(NDLEV)
      REAL*8              CRA(NDLEV,NDLEV),         CRCE(NDLEV,NDLEV)
      REAL*8              CRCP(NDLEV,NDLEV),        DENE,        DENP
      REAL*8              PCC(NDLEV,NDLEV)
```

3.46 b8norm: Subroutine b8norm from library adas2xx

```
C
      SUBROUTINE B8NORM( NDLEV  , NDMET  ,
     &                   NORD   ,
     &                   STCK   ,
     &                   PLAX   , PLX    ,
     &                   PLASX  , PLSX
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: B6STOT *********************
C
C  PURPOSE: TO NORMALISE TOTAL/SPECIFIC LINE POWERS FOR LEVEL 1
C           AND TOTAL EQUILIBRIUM LINE POWERS TO STAGE TOTAL POPULATION.
C
C  CALLING PROGRAM:  ADAS206
C
C  SUBROUTINE:
C
C  INPUT :  (I*4)  NDLEV   = MAXIMUM NUMBER OF ENERGY LEVELS ALLOWED
C  INPUT :  (I*4)  NDMET   = MAXIMUM NUMBER OF METASTABLE LEVELS ALLOWED
C
C  INPUT :  (I*4)  NORD    = NUMBER OF ORDINARY EXCITED LEVELS
C
C  INPUT :  (R*4)  STCK(,) = POPULATION MATRIX COVERING ALL NON-METAST-
C                            ABLE/ORDINARY EXCITED LEVELS AS FUNCTION
C                            OF METASTABLE INDEX.
C                            VALUES FOR GIVEN TEMPERATURE AND DENSITY.
C                            1st DIMENSION: ORDINARY EXCITED LEVEL INDEX
C                            2nd DIMENSION: METASTABLE LEVEL INDEX
C
C  I/O   :  (R*8)  PLAX    = INPUT:
C                            TOTAL EQUILIBRIUM LINE POWER COEFFICIENTS.
C                            AT FIXED TEMPERATURE AND DENSITY.
C                            (UNITS: ERGS CM3 SEC-1)
C                            OUTPUT:
C                            NORMALISED TO TOTAL STAGE POPULATION
C  I/O   :  (R*8)  PLX     = INPUT:
C                            TOTAL LINE POWERS FOR LEVEL 1 AT FIXED
C                            TEMPERATURE AND DENSITY.
C                            (UNITS: ERGS SEC-1).
C                            OUTPUT:
C                            NORMALISED TO TOTAL STAGE POPULATION
C
C  I/O   :  (R*8)  PLASX   = INPUT:
C                            SPECIFIC EQULIBRIUM LINE PWR COEFFICIENTS.
C                            AT FIXED TEMPERATURE AND DENSITY.
C                            (UNITS: ERGS CM3 SEC-1)
C                            OUTPUT:
C                            NORMALISED TO TOTAL STAGE POPULATION
C  I/O   :  (R*8)  PLSX    = INPUT:
C                            SPECIFIC LINE PWR FOR LEVEL 1 AT FIXED
C                            TEMPERATURE AND DENSITY.
C                            (UNITS: ERGS SEC-1).
C                            OUTPUT:
C                            NORMALISED TO TOTAL STAGE POPULATION
C
C           (I*4)  IS1      = ORDINARY EXCITED LEVEL INDEX
C
C           (R*8)  STOTX    = VARIABLE USED TO SUM STAGE TOTAL POPULATN.
```

```
C                            (INITIAL VALUE = 1 => GROUND)
C
C ROUTINES: NONE
C
C NOTE:
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/37
C          JET EXT. 5023
C
C DATE:    18/05/93
C
C UPDATE:  20/05/93 - P BRIDEN: STCK ARRAY CHANGED FROM REAL*8 -> REAL*4
C
C UNIX-IDL PORT:
C
C DATE: UNKNOWN
C
C AUTHOR: DAVID H BROOKS, UNIVERSITY OF STRATHCLYDE
C
C**********************************************************************
C PUT UNDER SCCS CONTROL:
C
C VERSION: 1.1 DATE: 10/05/96
C MODIFIED: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C     - FIRST PUT UNDER SCCS
C
C----------------------------------------------------------------------
C
C----------------------------------------------------------------------
      INTEGER             NDLEV,       NDMET,       NORD
      REAL*8              PLASX,       PLAX,        PLSX,        PLX
      REAL                STCK(NDLEV,NDMET)
```

3.47 b8popm: Subroutine b8popm from library adas2xx

```
C
      SUBROUTINE B8POPM( NDTEM  , NDDEN , NDMET , NDLEV ,
     &                   NPL    , NPLR  , NPLI  ,
     &                   MAXT   , MAXD  , NMET  ,
     &                            DENSA , IMETR ,
     &                            LRSEL , LISEL , LHSEL ,
     &                            RATPIA, RATMIA, RATHA ,
     &                   STCKM  , STVRM , STVIM , STVHM ,
     &                   POPAR
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: B8POPM *********************
C
C  PURPOSE: TO CONSTRUCT METASTABLE LEVEL POPULATIONS.
C
C  CALLING PROGRAM:  ADAS205/ADAS206
C
C  SUBROUTINE:
C
C  INPUT :  (I*4)  NDTEM   = MAXIMUM NUMBER OF TEMPERATURES ALLOWED
C  INPUT :  (I*4)  NDDEN   = MAXIMUM NUMBER OF DENSITIES ALLOWED
C  INPUT :  (I*4)  NDMET   = MAXIMUM NUMBER OF METASTABLE LEVELS ALLOWED
C  INPUT :  (I*4)  NDLEV   = MAXIMUM NUMBER OF ENERGY LEVELS ALLOWED
C  INPUT :  (I*4)  NPL     = NO. OF METASTABLES OF (Z+1) ION ACCESSED
C                            BY EXCITED STATE IONISATION IN COPASE
C                            FILE WITH IONISATION POTENTIALS GIVEN
C                            ON THE FIRST DATA LINE
C  INPUT :  (I*4)  NPLR    = NO. OF ACTIVE METASTABLES OF (Z+1) ION
C  INPUT :  (I*4)  NPLI    = NO. OF ACTIVE METASTABLES OF (Z-1) ION
C
C  INPUT :  (I*4)  MAXT    = NUMBER OF INPUT TEMPERATURES ( 1 ->'NDTEM')
C  INPUT :  (I*4)  MAXD    = NUMBER OF INPUT DENSITIES ( 1 ->'NDDEN')
C  INPUT :  (I*4)  NMET    = NUMBER OF METASTABLES LEVELS ( 1 ->'NDMET')
C
C  INPUT :  (R*8)  DENSA() = ELECTRON DENSITIES  (UNITS: CM-3)
C  INPUT :  (I*4)  IMETR() = INDEX OF METASTABLE IN COMPLETE LEVEL LIST
C                            (ARRAY SIZE = 'NDMET' )
C
C  INPUT :  (L*4)  LRSEL   = .TRUE.  => FREE ELECTRON RECOMBINATION
C                                       REQUESTED.
C                          = .FALSE. => FREE ELECTRON RECOMBINATION
C                                       NOT REQUESTED.
C  INPUT :  (L*4)  LHSEL   = .TRUE.  => CHARGE TRANSFER FROM NEUTRAL
C                                       HYDROGREN REQUESTED.
C                          = .FALSE. => CHARGE TRANSFER FROM NEUTRAL
C                                       HYDROGREN NOT REQUESTED.
C  INPUT :  (L*4)  LISEL   = .TRUE.  => IONISATION FROM LOWER IONIS.
C                                       STAGE REQUESTED.
C                          = .FALSE. => IONISATION FROM LOWER IONIS.
C                                       STAGE NOT REQUESTED.
C
C  INPUT :  (R*8)  RATPIA(,)= RATIO ( N(Z+1)/N(Z)  STAGE ABUNDANCIES )
C                             1ST DIMENSION: TEMP/DENS INDEX
C                             2ND DIMENSION: PARENT INDEX
C  INPUT :  (R*8)  RATMIA(,)= RATIO ( N(Z-1)/N(Z)  STAGE ABUNDANCIES )
C                             1ST DIMENSION: TEMP/DENS INDEX
C                             2ND DIMENSION: PARENT INDEX
C  INPUT :  (R*8)  RATHA() = RATIO (NEUTRAL H DENSITY/ELECTRON DENSITY)
```

```
C
C  INPUT :  (R*8)  STCKM(,,)= METASTABLE POPULATIONS STACK:
C                               1st DIMENSION: METASTABLE INDEX
C                               2nd DIMENSION: TEMPERATURE INDEX
C                               3rd DIMENSION: DENSITY INDEX
C  INPUT :  (R*8)  STVRM(,,,)= METASTABLE LEVEL:
C                            FREE-ELECTRON RECOMBINATION COEFFICIENTS
C                            (UNITS* CM**3/SEC-1)
C                              1ST DIMENSION: METASTABLE INDEX
C                              2ND DIMENSION: TEMPERATURE INDEX
C                              3RD DIMENSION: DENSITY INDEX
C                              4TH DIMENSION: PARENT INDEX
C  INPUT :  (R*8)  STVIM(,,,)= METASTABLE LEVEL:
C                            ELECTRON IMPACT IONISATION  COEFFICIENTS
C                            (UNITS* CM**3/SEC-1)
C                              1ST DIMENSION: METASTABLE INDEX
C                              2ND DIMENSION: TEMPERATURE INDEX
C                              3RD DIMENSION: DENSITY INDEX
C                              4TH DIMENSION: PARENT INDEX
C  INPUT :  (R*8)  STVHM(,,,)= METASTABLE LEVEL:
C                            CHARGE-EXCHANGE RECOMBINATION COEFFICIENTS
C                            (UNITS* CM**3/SEC-1)
C                             1st DIMENSION: METASTABLE INDEX
C                             2nd DIMENSION: TEMPERATURE INDEX
C                             3rd DIMENSION: DENSITY INDEX
C                             4TH DIMENSION: PARENT INDEX
C
C  OUTPUT:  (R*8)  POPAR(,,)= LEVEL POPULATIONS
C                            1ST DIMENSION: LEVEL INDEX
C                            2ND DIMENSION: TEMPERATURE INDEX
C                            3RD DIMENSION: DENSITY INDEX
C                            (ON OUTPUT CONTAINS POPULATIONS FOR
C                             METASTABLE LEVELS ONLY.)
C
C           (R*8) DCOEF     = DENSITY MULTIPLIED BY RELEVANT RATIOS  FOR
C                             CALCULATING RECOMBINATION CONTRIBUTIONS.
C
C           (I*4) IT        = TEMPERATURE ARRAY INDEX
C           (I*4) IP        = PARENT INDEX
C           (I*4) IN        = DENSITY ARRAY INDEX
C           (I*4) IM        = METASTABLE LEVEL ARRAY INDEX
C
C
C ROUTINES: NONE
C
C AUTHOR:  HP SUMMERS (UPGRADE OF BXPOPM BY PE BRIDEN)
C          K1/1/57
C          JET EXT. 4941
C
C DATE:    11/06/92
C
C-----------------------------------------------------------------------
C UNIX-IDL PORT:
C
C AUTHOR: DAVID H BROOKS, UNIVERSITY OF STRATHCLYDE
C
C DATE: UNKNOWN
C
C-----------------------------------------------------------------------
C PUT UNDER SCCS CONTROL:
C
```

```
C VERSION: 1.1 DATE: 10/05/96
C MODIFIED: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C     - FIRST PUT UNDER SCCS
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IMETR(NDMET),              MAXD,         MAXT
      INTEGER             NDDEN,       NDLEV,        NDMET,        NDTEM
      INTEGER             NMET,        NPL,          NPLI,         NPLR
      LOGICAL             LHSEL,       LISEL,        LRSEL
      REAL*8              DENSA(NDDEN)
      REAL*8              POPAR(NDLEV,NDTEM,NDDEN), RATHA(NDDEN)
      REAL*8              RATMIA(NDDEN,NDMET),      RATPIA(NDDEN,NDMET)
      REAL*8              STCKM(NDMET,NDTEM,NDDEN)
      REAL*8              STVHM(NDMET,NDTEM,NDDEN,NDMET)
      REAL*8              STVIM(NDMET,NDTEM,NDDEN,NDMET)
      REAL*8              STVRM(NDMET,NDTEM,NDDEN,NDMET)
```

3.48 b8popo: Subroutine b8popo from library adas2xx

```
C
      SUBROUTINE B8POPO( NDTEM  , NDDEN , NDMET , NDLEV ,
     &                   NPL    , NPLR  , NPLI  ,
     &                   MAXT   , MAXD  , NMET  , NORD  ,
     &                            DENSA , IMETR , IORDR ,
     &                            LRSEL , LISEL , LHSEL ,
     &                            RATPIA, RATMIA, RATHA ,
     &                   STACK  , STVR  , STVI  , STVH  ,
     &                   POPAR
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: B8POPO *********************
C
C  PURPOSE: TO CONSTRUCT ORDINARY/NON-METASTABLE LEVEL POPULATIONS.
C
C  CALLING PROGRAM:  ADAS205/ADAS206
C
C  SUBROUTINE:
C
C  INPUT :  (I*4)  NDTEM   = MAXIMUM NUMBER OF TEMPERATURES ALLOWED
C  INPUT :  (I*4)  NDDEN   = MAXIMUM NUMBER OF DENSITIES ALLOWED
C  INPUT :  (I*4)  NDMET   = MAXIMUM NUMBER OF METASTABLE LEVELS ALLOWED
C  INPUT :  (I*4)  NDLEV   = MAXIMUM NUMBER OF ENERGY LEVELS ALLOWED
C  INPUT :  (I*4)  NPL     = NO. OF METASTABLES OF (Z+1) ION ACCESSED
C                            BY EXCITED STATE IONISATION IN COPASE
C                            FILE WITH IONISATION POTENTIALS GIVEN
C                            ON THE FIRST DATA LINE
C  INPUT :  (I*4)  NPLR    = NO. OF ACTIVE METASTABLES OF (Z+1) ION
C  INPUT :  (I*4)  NPLI    = NO. OF ACTIVE METASTABLES OF (Z-1) ION
C
C  INPUT :  (I*4)  MAXT    = NUMBER OF INPUT TEMPERATURES ( 1 ->'NDTEM')
C  INPUT :  (I*4)  MAXD    = NUMBER OF INPUT DENSITIES ( 1 ->'NDDEN')
C  INPUT :  (I*4)  NMET    = NUMBER OF METASTABLES LEVELS ( 1 ->'NDMET')
C  INPUT :  (I*4)  NORD    = NUMBER OF ORDINARY LEVELS ( 1 ->'NDLEV')
C
C  INPUT :  (R*8)  DENSA() = ELECTRON DENSITIES  (UNITS: CM-3)
C  INPUT :  (I*4)  IMETR() = INDEX OF METASTABLE IN COMPLETE LEVEL LIST
C                            (ARRAY SIZE = 'NDMET' )
C  INPUT :  (I*4)  IORDR() =INDEX OF ORDINARY EXCITED LEVELS IN COMPLETE
C                           LEVEL LIST.
C                           (ARRAY SIZE = 'NDLEV' )
C
C  INPUT :  (L*4)  LRSEL   = .TRUE.  => FREE ELECTRON RECOMBINATION
C                                       REQUESTED.
C                          = .FALSE. => FREE ELECTRON RECOMBINATION
C                                       NOT REQUESTED.
C  INPUT :  (L*4)  LISEL   = .TRUE.  => ELECTRON IMPACT IONISATION
C                                       REQUESTED.
C                          = .FALSE. => ELECTRON IMPACT IONISATION
C                                       NOT REQUESTED.
C  INPUT :  (L*4)  LHSEL   = .TRUE.  => CHARGE TRANSFER FROM NEUTRAL
C                                       HYDROGREN REQUESTED.
C                          = .FALSE. => CHARGE TRANSFER FROM NEUTRAL
C                                       HYDROGREN NOT REQUESTED.
C
C  INPUT :  (R*8)  RATPIA(,)= RATIO ( N(Z+1)/N(Z)  STAGE ABUNDANCIES )
C                             1ST DIMENSION: TEMP/DENS INDEX
C                             2ND DIMENSION: PARENT INDEX
```

```
C  INPUT :  (R*8)  RATMIA(,)= RATIO ( N(Z-1)/N(Z)  STAGE ABUNDANCIES )
C                             1ST DIMENSION: TEMP/DENS INDEX
C                             2ND DIMENSION: PARENT INDEX
C  INPUT :  (R*8)  RATHA() = RATIO (NEUTRAL H DENSITY/ELECTRON DENSITY)
C
C  INPUT : (R*4) STACK(,,,)= ORDINARY EXCITED LEVEL POPULAT'N DEPENDENCE
C                           ON METASTABLE LEVEL.
C                            1st DIMENSION: ORDINARY LEVEL INDEX
C                            2nd DIMENSION: METASTABLE INDEX
C                            3rd DIMENSION: TEMPERATURE INDEX
C                            4th DIMENSION: DENSITY INDEX
C  INPUT : (R*4) STVR(,,,) = ORDINARY EXCITED LEVEL:
C                            FREE-ELECTRON RECOMBINATION COEFFICIENTS
C                            (UNITS* CM**3/SEC-1)
C                            1st DIMENSION: ORDINARY LEVEL INDEX
C                            2nd DIMENSION: TEMPERATURE INDEX
C                            3rd DIMENSION: DENSITY INDEX
C                            4TH DIMENSION: PARENT INDEX
C  INPUT : (R*4) STVI(,,,) = ORDINARY EXCITED LEVEL:
C                            ELECTRON IMPACT IONISATION  COEFFICIENTS
C                            (UNITS* CM**3/SEC-1)
C                            1st DIMENSION: ORDINARY LEVEL INDEX
C                            2nd DIMENSION: TEMPERATURE INDEX
C                            3rd DIMENSION: DENSITY INDEX
C                            4TH DIMENSION: PARENT INDEX
C  INPUT : (R*4) STVH(,,,) = ORDINARY EXCITED LEVEL:
C                            CHARGE-EXCHANGE RECOMBINATION COEFFICIENTS
C                            (UNITS* CM**3/SEC-1)
C                            1st DIMENSION: ORDINARY LEVEL INDEX
C                            2nd DIMENSION: TEMPERATURE INDEX
C                            3rd DIMENSION: DENSITY INDEX
C                            4TH DIMENSION: PARENT INDEX
C
C  I/O   :  (R*8)  POPAR(,,)= LEVEL POPULATIONS
C                             1st DIMENSION: LEVEL INDEX
C                             2nd DIMENSION: TEMPERATURE INDEX
C                             3rd DIMENSION: DENSITY INDEX
C                              ON INPUT : CONTAINS POPULATIONS FOR
C                                         METASTABLE LEVELS ONLY.
C                              ON OUTPUT: CONTAINS POPULATIONS FOR
C                                         ALL LEVELS.
C
C           (R*8) DCOEF     = DENSITY MULTIPLIED BY RELEVANT RATIOS  FOR
C                             CALCULATING RECOMBINATION CONTRIBUTIONS.
C
C           (I*4) IT        = TEMPERATURE ARRAY INDEX
C           (I*4) IP        = PARENT INDEX
C           (I*4) IN        = DENSITY ARRAY INDEX
C           (I*4) IO        = ORDINARY LEVEL ARRAY INDEX
C           (I*4) IM        = METASTABLE LEVEL ARRAY INDEX
C
C
C ROUTINES: NONE
C
C AUTHOR:  HP SUMMERS  (UPDATE OF BXPOPO BY PE BRIDEN)
C          K1/1/57
C          JET EXT. 4941
C
C DATE:    11/06/92
C
C UPDATE:  12/07/93  HPS - CHANGE STSCK, STVR, STVI, STVH
```

```
C                           DIMENSIONS TO R*4
C*********************************************************************
C UNIX-IDL PORT:
C
C AUTHOR: DAVID H BROOKS, UNIVERSITY OF STRATHCLYDE
C
C DATE: UNKNOWN
C
C*********************************************************************
C PUT UNDER SCCS CONTROL:
C
C VERSION: 1.1 DATE: 10/05/96
C MODIFIED: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C     - FIRST PUT UNDER SCCS
C
C---------------------------------------------------------------------
C
C---------------------------------------------------------------------
      INTEGER             IMETR(NDMET),              IORDR(NDLEV)
      INTEGER             MAXD,         MAXT,        NDDEN,       NDLEV
      INTEGER             NDMET,        NDTEM,       NMET,        NORD
      INTEGER             NPL,          NPLI,        NPLR
      LOGICAL             LHSEL,        LISEL,       LRSEL
      REAL*8              DENSA(NDDEN)
      REAL*8              POPAR(NDLEV,NDTEM,NDDEN), RATHA(NDDEN)
      REAL*8              RATMIA(NDDEN,NDMET),      RATPIA(NDDEN,NDMET)
      REAL                STACK(NDLEV,NDMET,NDTEM,NDDEN)
      REAL                STVH(NDLEV,NDTEM,NDDEN,NDMET)
      REAL                STVI(NDLEV,NDTEM,NDDEN,NDMET)
      REAL                STVR(NDLEV,NDTEM,NDDEN,NDMET)
```

3.49 b8rcom: Subroutine b8rcom from library adas2xx

```
C
      SUBROUTINE B8RCOM( NDTEM  , NDTRN  , NDLEV  , NDMET   ,
     &                   NTIN   , TIN    , RCIN   ,
     &                   NTOUT  , TOUT   ,
     &                   ICNT   , ITRN   , ICLEV  , IC2LEV  ,
     &                   RCOUT  , LTRNG
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: B8RCOM *********************
C
C  PURPOSE: TO ESTABLISH RECOMBINATION RATE COEFFICIENTS  FOR  A SET  OF
C           TEMPERATURES GIVEN BY THE ARRAY 'TOUT()' USING CUBIC SPLINES
C           ON  A SET OF RATE  COEFFICIENTS  COVERING  THE  TEMPERATURES
C           GIVEN BY THE ARRAY 'TIN()'.
C
C           RECOMBINATION TYPE IS SELECTED VIA 'ICNT' & 'ITRN'
C
C           RATE COEFFICIENTS ARE GIVEN FOR A NUMBER OF CAPTURING LEVELS
C           AND THE ARRAY 'RCOUT(,,)' REPRESENTS  COEFFTS.  FOR COMB-
C           INATIONS OF TEMPERATURE, CAPTURING LEVEL INDEX AND PARENT
C           INDEX.
C
C           SPLINE IS CARRIED OUT USING LOG(RATE COEFFICIENT VALUES)
C
C  CALLING PROGRAM:  ADAS205/ADAS206
C
C  SUBROUTINE:
C
C  INPUT :  (I*4)  NDTEM   = MAXIMUM NUMBER OF TEMPERATURES ALLOWED
C  INPUT :  (I*4)  NDTRN   = MAXIMUM NUMBER OF RECOMBINATIONS ALLOWED
C  INPUT :  (I*4)  NDLEV   = MAXIMUM NUMBER OF ENERGY LEVELS ALLOWED
C  INPUT :  (I*4)  NDMET   = MAXIMUM NUMBER OF METASTABLES ALLOWED
C
C  INPUT :  (I*4)  NTIN    = NUMBER OF TEMPERATURES REPRESENTED IN THE
C                            INPUT DATA SET.
C  INPUT :  (R*8)  TIN()   = TEMPERATURES REPRESENTED IN INPUT DATA SET
C  INPUT :  (R*8)  RCIN(,) = RATE COEFF. REPRESENTED IN INPUT DATA SET
C                            1st DIMENSION: TEMPERATURE INDEX ('TIN')
C                            2nd DIMENSION: RECOMBINATION INDEX
C                                           (SEE: 'ITRN()')
C
C  INPUT :  (I*4)  NTOUT   = NUMBER OF ISPF SELECTED TEMPERATURES FOR
C                            OUTPUT.
C  INPUT :  (R*8)  TOUT()  = ISPF SELECTED TEMPERATURES FOR OUTPUT.
C
C  INPUT :  (I*4)  ICNT    = NUMBER OF SELECTED RECOMBINATIONS
C  INPUT :  (I*4)  ITRN()  = INDEX VALUES IN MAIN TRANSITION ARRAY WHICH
C                            REPRESENT RECOMBINATIONS  OF THE  SELECTED
C                            TYPE
C                            USED TO SELECT APPROPRIATE RATE COEFFTS FOR
C                            RECOMBINATION TYPE.
C  INPUT :  (I*4)  ICLEV() = CAPTURING LEVELS INDICES.
C                            DIMENSION: 'TRANSITION'/RECOMBINATION INDEX
C  INPUT :  (I*4)  IC2LEV()= PARENT INDEX.
C                            DIMENSION: 'TRANSITION'/RECOMB/IONIS  INDEX
C
C  OUTPUT:  (R*8)  RCOUT(,,)= SPLINED RECOMBINATION RATE COEFFT. VALUES.
C                            1st DIMENSION: TEMPERATURE INDEX ('TOUT')
```

```
C                             2nd DIMENSION: CAPTURING LEVEL INDEX.
C                             3RD DIMENSION: PARENT INDEX.
C
C  OUTPUT:  (L*4)  LTRNG() = .TRUE. => TEMPERATURE VALUES WITHIN RANGE
C                                      READ FROM INPUT COPASE DATA SET.
C                           = .FALSE.=>TEMPERATURE VALUE NOT WITHIN RANGE
C                                     READ FROM INPUT COPASE DATA SET.
C                             1st DIMENSION: TEMPERATURE INDEX.
C
C
C           (I*4)  NTDSN   = PARAMETER = MAXIMUM NUMBER OF TEMPERATURES
C                                        ALLOWED IN INPUT DATA SET = 14
C           (I*4)  NLTEM   = PARAMETER = MUST BE >= 'NDTEM'
C
C           (I*4)  IOPT    = SPLINE END CONDITIONS/EXTRAPOLATION CONTROL
C                            SWITCH - SEE 'XXSPLE'
C                            I.E. DEFINES THE BOUNDARY DERIVATIVES.
C                            (VALID VALUES = 0, 1, 2, 3, 4)
C           (I*4)  IRECMB  = APPROPRIATE RECOMBINATN INDEX FOR 'RCIN(,)'
C           (I*4)  ICAP    = CAPTURING LEVEL INDEX BEING ASSESSED.
C           (I*4)  IC      = RECOMBINATION ARRAY INDEX
C           (I*4)  IP      = PARENT INDEX
C           (I*4)  IT      = TEMPERATURE ARRAY INDEX
C
C           (R*8)  DYIN()  = INTERPOLATED DERIVATIVES
C                            DIMENSION: TEMPERATURE INDEX ('TIN()')
C
C           (L*4)  LSETX   = .TRUE.  => X-AXES ('TIN()' VALUES) NEED TO
C                                       SET IN 'XXSPLE'.
C                            .FALSE. => X-AXES ('TIN()' VALUES) HAVE
C                                       BEEN SET IN 'XXSPLE'.
C                            (NOTE: 'LSETX' IS RESET BY 'XXSPLE')
C
C           (R*8)  LRCIN() = LOG ( 'RCIN(,)' ) FOR GIVEN CAPTURING LEVEL
C                            DIMENSION: TEMPERATURE INDEX ('TIN()')
C           (R*8)  LRCOUT()= LOG ( SPLINED RECOMB.IONIS  RATE COEFTS )
C                            DIMENSION: TEMPERATURE INDEX ('TOUT()' )
C
C
C ROUTINES:
C           ROUTINE    SOURCE    BRIEF DESCRIPTION
C           ------------------------------------------------------------
C           XXSPLE     ADAS      SPLINE SUBROUTINE (WITH EXTRAP. INFO)
C
C
C AUTHOR:  HP SUMMERS (UPGRADE OF BXRCOM BY PE BRIDEN)
C          K1/1/57
C          JET EXT. 4941
C
C DATE:    11/06/92
C
C UPDATE:  12/07/93  HPS - MODIFICATIONS TO MAKE CONSISTENT WITH
C                          LATEST VERSION OF B8DATA
C
C-----------------------------------------------------------------------
C UNIX-IDL PORT:
C
C AUTHOR: DAVID H BROOKS, UNIVERSITY OF STRATHCLYDE
C
C DATE: UNKNOWN
C
```

```
C*********************************************************************
C PUT UNDER SCCS CONTROL:
C
C VERSION: 1.1 DATE: 10/05/96
C MODIFIED: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C     - FIRST PUT UNDER SCCS
C
C---------------------------------------------------------------------
C
C---------------------------------------------------------------------
      INTEGER             IC2LEV(NDTRN),             ICLEV(NDTRN)
      INTEGER             ICNT,         ITRN(NDTRN), NDLEV,       NDMET
      INTEGER             NDTEM,        NDTRN,       NTIN,        NTOUT
      LOGICAL             LTRNG(NDTEM)
      REAL*8              RCIN(NTDSN,NDTRN)
      REAL*8              RCOUT(NDTEM,NDLEV,NDMET), TIN(NTDSN)
      REAL*8              TOUT(NDTEM)
```

3.50 b8scom: Subroutine b8scom from library adas2xx

```
C
      SUBROUTINE B8SCOM( NDTEM , NDTRN      , NDLEV  , NDMET  ,
     &                   IL    , WA         , NPL    , BWNOA  ,
     &                   NMET  , IMETR      , NORD   , IORDR  ,
     &                   NV    , SCEF       , SCOM   ,
     &                   MAXT  , TEA        ,
     &                   ICNTS , ISTRN      , IS1A   , IS2A   ,
     &                   LSSETA, SGRDA      , ESGRDA ,
     &                   SMETA , ESMETA     , SORDA  , ESORDA ,
     &                   LTRNG
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: B8SCOM *********************
C
C  PURPOSE: TO ESTABLISH IONISATION RATE COEFFICIENTS  Z --> Z+1 FOR  A
C           SET  OF TEMPERATURES GIVEN BY THE ARRAY 'TEA()' USING CUBIC
C           SPLINES ON  A SET OF RATE  COEFFICIENTS  COVERING  THE
C           TEMPERATURES GIVEN BY THE ARRAY 'SCEF()'.
C
C           IONISATION DATA COMES EITHER FROM AN INTERACTIVE SEARCH VIA
C           THE ADAS208/ADAS502 ROUTE OR DIRECTLY FROM THE INPUT ADF04
C           FILE.
C
C           THE OUTPUT IS SEPARATED INTO THE METASTABLE PART (SMETA)AND
C           THE ORDINARY LEVEL PART (SORDA) APPROPRIATELY INDEXED.
C           EXPONENTIAL FACTORS (ESMETA AND (ESORDA) ARE KEPT SEPARATE
C           FROM THE REMAINDER OF THE RATE COEFFICIENTS.
C
C           IONISATION TYPE IS SELECTED VIA 'ICNTS' & 'ISTRN'
C
C           RATE COEFFICIENTS ARE GIVEN FOR A NUMBER OF IONISING LEVELS
C           AND THE ARRAY 'SGRDA(,,)' REPRESENTS  COEFFTS.  FOR COMB-
C           INATIONS OF TEMPERATURE, IONISING LEVEL INDEX AND FINAL
C           PARENT INDEX.
C
C           SPLINE IS CARRIED OUT USING LOG(RATE COEFFICIENT VALUES)
C
C  CALLING PROGRAM:  ADAS208
C
C  SUBROUTINE:
C
C  INPUT :  (I*4)  NDTEM   = MAXIMUM NUMBER OF TEMPERATURES ALLOWED
C  INPUT :  (I*4)  NDTRN   = MAXIMUM NUMBER OF TRANSITIONS ALLOWED
C  INPUT :  (I*4)  NDLEV   = MAXIMUM NUMBER OF ENERGY LEVELS ALLOWED
C  INPUT :  (I*4)  NDMET   = MAXIMUM NUMBER OF METASTABLES ALLOWED

C  INPUT :  (I*4)  IL      = NUMBER OF ENERGY LEVELS
C  INPUT :  (R*8)  WA()    = ENERGY LEVELS RELATIVE TO LOWEST(CM-1)
C  INPUT :  (I*4)  NPL     = NUMBER OF PARENTS
C  INPUT :  (R*8)  BWNOA() = PARENT ENERGIES RELATIVE TO RECOMBINED
C                            ION GROUND LEVEL (CM-1)
C
C  INPUT :  (I*4)  NMET    = NUMBER OF RECOMBINED METASTABLES
C  INPUT :  (I*4)  IMETR() = INDICES OF METASTABLES IN FULL LEVEL LIST
C  INPUT :  (I*4)  NORD    = NUMBRE OF ORDINARY EXCITED LEVELS
C  INPUT :  (I*4)  IORDR() = INDICES OF ORDINARY LEVELS IN FULL LEVEL LIST
C
C  INPUT :  (I*4)  NV      = NUMBER OF TEMPERATURES REPRESENTED IN THE
```

```
C                              INPUT DATA SET.
C  INPUT :  (R*8)  SCEF()  = TEMPERATURES REPRESENTED IN INPUT DATA SET
C  INPUT :  (R*8)  SCOM(,)= RATE COEFF. REPRESENTED IN INPUT DATA SET
C                              1st DIMENSION: TEMPERATURE INDEX ('SCEF')
C                              2nd DIMENSION: IONISATION INDEX
C                                             (SEE: 'ISTRN()')
C
C  INPUT :  (I*4)  MAXT    = NUMBER OF ISPF SELECTED TEMPERATURES FOR
C                              OUTPUT.
C  INPUT :  (R*8)  TEA()   = ISPF SELECTED TEMPERATURES FOR OUTPUT.
C
C  INPUT :  (I*4)  ICNTS   = NUMBER OF SELECTED IONISATIONS
C  INPUT :  (I*4)  ISTRN() = INDEX VALUES IN MAIN TRANSITION ARRAY WHICH
C                              REPRESENT IONISATIONS  OF THE  SELECTED
C                              TYPE - USED TO SELECT APPROPRIATE RATE COEFFTS
C                              FOR IONISATION Z --> Z+1 TYPE.
C  INPUT :  (I*4)  IS1A(  )= PARENT INDEX.
C                              DIMENSION: 'TRANSITION'/IONISATION  INDEX
C  INPUT :  (I*4)  IS2A()  = IONISING LEVELS INDICES.
C                              DIMENSION: 'TRANSITION'/IONISATION INDEX
C
C  INPUT :  (I*4)  LSSETA(,)=.TRUE.  => IONISATION DATA FROM ADAS502 ROUTE
C                             .FALSE. => NOT AVAILABLE FROM ADAS502 ROUTE
C                              1ST DIM: METASTABLE INDEX FROM MET. LIST
C                              2ND DIM: PARENT INDEX
C  INPUT :  (R*8)  SGRDA(,,)= INPUT IONISATION RATE COEFFT. VALUES.
C                              FROM THE ADAS208/ADAS502 LOOP
C                              (EXCLUDING EXPONENTIAL TEMPERATURE FACTOR)
C                             1st DIMENSION: TEMPERATURE INDEX ('TOUT')
C                             2nd DIMENSION: IONISING LEVEL INDEX.
C                             3RD DIMENSION: PARENT INDEX.
C  INPUT :  (R*8)  ESGRDA(,,)= IONISATION RATE COEFFT. EXPONENTIAL FACTORS
C                               FROM THE ADAS208/ADAS502 LOOP
C                             1st DIMENSION: TEMPERATURE INDEX ('TOUT')
C                             2nd DIMENSION: IONISING LEVEL INDEX.
C                             3RD DIMENSION: PARENT INDEX.
C
C  OUTPUT:  (R*8)  SMETA(,,)= SPLINED IONISATION RATE COEFFT. VALUES.
C                              FOR THE METASTABLES
C                              (EXCLUDING EXPONENTIAL TEMPERATURE FACTOR)
C                             1st DIMENSION: TEMPERATURE INDEX ('TOUT')
C                             2nd DIMENSION: IONISING METASTABLE INDEX.
C                             3RD DIMENSION: PARENT INDEX.
C  OUTPUT:  (R*8)  ESMETA(,,)= SPLINED IONISATION RATE COEFFT.
C                               EXPONENTIAL TEMPERATURE FACTORS.
C                             1st DIMENSION: TEMPERATURE INDEX ('TOUT')
C                             2nd DIMENSION: IONISING METASTABLE INDEX.
C                             3RD DIMENSION: PARENT INDEX.
C  OUTPUT:  (R*8)  SORDA(,,)= SPLINED IONISATION RATE COEFFT. VALUES.
C                              FOR THE ORDINARY LEVELS
C                              (EXCLUDING EXPONENTIAL TEMPERATURE FACTOR)
C                             1st DIMENSION: TEMPERATURE INDEX ('TOUT')
C                             2nd DIMENSION: IONISING ORDINARY LEVEL INDEX.
C                             3RD DIMENSION: PARENT INDEX.
C  OUTPUT:  (R*8)  ESORDA(,,)= SPLINED IONISATION RATE COEFFT.
C                               EXPONENTIAL TEMPERATURE FACTORS.
C                             1st DIMENSION: TEMPERATURE INDEX ('TOUT')
C                             2nd DIMENSION: IONISING ORDINARY LEVEL INDEX.
C                             3RD DIMENSION: PARENT INDEX.
C
C  OUTPUT:  (L*4)  LTRNG() = .TRUE. => TEMPERATURE VALUES WITHIN RANGE
```

```
C                                     READ FROM INPUT COPASE DATA SET.
C                           = .FALSE.=>TEMPERATURE VALUE NOT WITHIN RANGE
C                                     READ FROM INPUT COPASE DATA SET.
C                             1st DIMENSION: TEMPERATURE INDEX.
C
C
C           (I*4)  NTDSN   = PARAMETER = MAXIMUM NUMBER OF TEMPERATURES
C                                        ALLOWED IN INPUT DATA SET = 14
C           (I*4)  NLTEM   = PARAMETER = MUST BE >= 'NDTEM'
C
C           (I*4)  IOPT    = SPLINE END CONDITIONS/EXTRAPOLATION CONTROL
C                            SWITCH - SEE 'XXSPLE'
C                            I.E. DEFINES THE BOUNDARY DERIVATIVES.
C                            (VALID VALUES = 0, 1, 2, 3, 4)
C           (I*4)  I       = GENERAL INDEX
C           (I*4)  ICAP    = CAPTURING LEVEL INDEX BEING ASSESSED.
C           (I*4)  IC      = RECOMBINATION ARRAY INDEX
C           (I*4)  IP      = PARENT INDEX
C           (I*4)  IT      = TEMPERATURE ARRAY INDEX
C
C           (R*8)  DYIN()  = INTERPOLATED DERIVATIVES
C                            DIMENSION: TEMPERATURE INDEX ('TIN()')
C
C           (L*4)  LSETX   = .TRUE.  => X-AXES ('TIN()' VALUES) NEED TO
C                                       SET IN 'XXSPLE'.
C                            .FALSE. => X-AXES ('TIN()' VALUES) HAVE
C                                       BEEN SET IN 'XXSPLE'.
C                            (NOTE: 'LSETX' IS RESET BY 'XXSPLE')
C
C           (R*8)  LSCOM() = LOG ( 'SCOM(,)' ) FOR GIVEN IONISING LEVEL
C                            DIMENSION: TEMPERATURE INDEX ('SCEF()')
C           (R*8)  LSGRD()= LOG ( SPLINED IONIS  RATE COEFTS )
C                            DIMENSION: TEMPERATURE INDEX ('TEA()' )
C
C
C ROUTINES:
C           ROUTINE    SOURCE    BRIEF DESCRIPTION
C           ------------------------------------------------------------
C           XXSPLE     ADAS      SPLINE SUBROUTINE (WITH EXTRAP. INFO)
C
C
C AUTHOR:  HP SUMMERS, UNIVERSITY OF STRATHCLYDE
C          TEL. 0141-548-4196
C
C DATE:    14/09/99
C
C
C VERSION  : 1.2
C DATE     : 19/12/2000
C MODIFIED : Martin O'Mullane
C             - Excluded S values GT 1.0 from the spline fit in order
C               to compensate for numerical problems at low temperatures.
C
C VERSION  : 1.3
C DATE     : 17/02/2006
C MODIFIED : Martin O'Mullane
C             - Te values for S-line splining may not be the same
C               so set lsetx to TRUE before call to xxsple.
C             - Set unused values in redscef and redlscom to 0.0.
C
C-----------------------------------------------------------------------
```

```
C
C-----------------------------------------------------------------------
      INTEGER             ICNTS,       IL,          IMETR(NDMET)
      INTEGER             IORDR(NDLEV),             IS1A(NDLEV)
      INTEGER             IS2A(NDLEV), ISTRN(NDTRN),             MAXT
      INTEGER             NDLEV,       NDMET,       NDTEM,       NDTRN
      INTEGER             NMET,        NORD,        NPL,         NV
      LOGICAL             LSSETA(NDMET,NDMET),      LTRNG(NDTEM)
      REAL*8              BWNOA(NDMET)
      REAL*8              ESGRDA(NDTEM,NDMET,NDMET)
      REAL*8              ESMETA(NDTEM,NDMET,NDMET)
      REAL*8              ESORDA(NDTEM,NDLEV,NDMET)
      REAL*8              SCEF(NDTEM), SCOM(NTDSN,NDTRN)
      REAL*8              SGRDA(NDTEM,NDMET,NDMET)
      REAL*8              SMETA(NDTEM,NDMET,NDMET)
      REAL*8              SORDA(NDTEM,NDLEV,NDMET), TEA(NDTEM)
      REAL*8              WA(NDLEV)
```

3.51 b8setp: Subroutine b8setp from library adas2xx

```
       SUBROUTINE B8SETP( IZ0    , IZ     ,
     &                    NDLEV  , IL     , ICNTE ,
     &                    CSTRGA , ISA    , ILA   , XJA  ,
     &                    STRGA  , NPL    , CPRTA , NDMET,
     &                    LSS04A ,
     &                    STRGMF , STRGMI
     &                  )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: B8SETP *********************
C
C  PURPOSE:  TO SET UP PARAMETERS IN THE SHARED POOLED FOR PANEL DISPLAY
C
C  CALLING PROGRAM: ADAS208
C
C  DATA:
C           DATA IS OBTAINED VIA SUBROUTINE 'BADATA'
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  IZ0     =         NUCLEAR CHARGE READ
C  INPUT : (I*4)  IZ      =  RECOMBINED ION CHARGE READ
C
C  INPUT : (I*4)  NDLEV   = MAXIMUM NUMBER OF ENERGY LEVELS ALLOWED
C  INPUT : (I*4)  IL      = INPUT DATA FILE: NUMBER OF ENERGY LEVELS
C  INPUT : (I*4)  ICNTE   = NUMBER OF ELECTRON IMPACT TRANSITIONS
C
C  INPUT : (C*18) CSTRGA()= NOMENCLATURE/CONFIGURATION FOR LEVEL 'IA()'
C  INPUT : (I*4)  ISA()   = MULTIPLICITY FOR LEVEL 'IA()'
C                           NOTE: (ISA-1)/2 = QUANTUM NUMBER (S)
C  INPUT : (I*4)  ILA()   = QUANTUM NUMBER (L) FOR LEVEL 'IA()'
C  INPUT : (R*8)  XJA()   = QUANTUM NUMBER (J-VALUE) FOR LEVEL 'IA()'
C                           NOTE: (2*XJA)+1 = STATISTICAL WEIGHT
C  INPUT : (I*4)  NPL     = NUMBER OF PARENTS IN INPUT DATA SET
C  INPUT : (C*9)  CPRTA() = PARENT NAME FROM INPUT DATA SET
C  INPUT : (I*4)  NDMET   = MAX.NO.OF METASTABLES ALLOWED
C  I/O   : (L*4)  LSS04A(,)= .TRUE. => IONIS. RATE SET IN ADF04 FILE:
C                            .FALSE.=> NOT SET IN ADF04 FILE
C                            1ST DIM: LEVEL INDEX
C                            2ND DIM: PARENT METASTABLE INDEX
C
C  OUTPUT: (C*22) STRGA() = LEVEL DESIGNATIONS
C  OUTPUT: (C*11) STRGMF()= INFORMATION STRINGS FOR IDL
C  OUTPUT: (C*12) STRGMI()= INFORMATION STRINGS FOR IDL
C
C          (C*8)  F6      = PARAMETER = 'VREPLACE'
C
C          (I*4)  ILEN    = LENGTH, IN BYTES, OF ISPF DIALOG VARIABLES
C          (I*4)  ILEV    = ARRAY COUNTER FOR LEVEL INDEX
C          (I*4)  J       = VALUE OF QUANTUM NUMBER L + 1
C          (I*4)  LFPOOL  = NO. OF LEVEL STRINGS SENT TO FUNCTION POOL
C
C          (C*2)  SZ0     =         NUCLEAR CHARGE READ
C          (C*2)  SZ      =  RECOMBINED ION CHARGE READ
C          (C*4)  SCNTE   =  NUMBER OF ELECTRON IMPACT TRANSITIONS
C          (C*4)  SIL     =  NUMBER OF ENERGY LEVELS
C          (C*1)  CONFIG()= QUANTUM NUMBER (L) LETTERS
C                           DIMENSION: QUANTUM NUMBER L + 1
```

```
C          (C*8)  CHA()    = FUNCTION POOL NAMES: CHARGE VALUES
C          (C*8)  CHB()    = FUNCTION POOL NAMES: LEVEL DESIGNATIONS <=99
C          (I*4)  PIPEOU   = PARAMETER = UNIT NUMBER FOR OUTPUT TO PIPE
C
C
C ROUTINES:
C           ROUTINE     SOURCE      BRIEF DESCRIPTION
C           ------------------------------------------------------------
C
C AUTHOR:  D.H.BROOKS, UNIVERSITY OF STRATHCLYDE
C
C DATE:    11/04/96
C
C VERSION: 1.0                             DATE: 11/04/96
C MODIFIED: DH BROOKS
C      - CREATED FROM SKELETON OF BXSETP. ADDED NPL, NDMET & CPRTA TO
C        PARAMETER LIST & SENT CPRTA THROUGH TO IDL AS STRGMF. ALSO
C        BUILT STRGMI TO SEND THROUGH FOR 502V208.
C
C VERSION: 1.1                             DATE: 10/05/96
C MODIFIED: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C      - CHANGED DIMENSIONING OF CPRTA TO AGREE WITH ADAS208 AND ALTERED
C        LOOP TO WRITE STRGMF SO ARRAY INDEX DIDN'T GO OUT OF BOUNDS.
C        PUT UNDER S.C.C.S.
C
C VERSION: 1.2                             DATE: 20/05/96
C MODIFIED: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C      - ADDED XXFLSH CALLS
C
C VERSION: 1.3 DATE: 13/09/99
C MODIFIED: HUGH SUMMERS, UNIVERSITY OF STRATHCLYDE
C     - ADDED TRUTH TABLE FOR S-LINE IONISATION DATA AVAILABILITY
C             IN ADF04 FILE TO INFORMATION TRANSFER TO IDL
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      CHARACTER*9         CPRTA(NDMET)
      CHARACTER*18        CSTRGA(IL)
      CHARACTER*22        STRGA(NDLEV)
      CHARACTER*11        STRGMF(NDMET)
      CHARACTER*12        STRGMI(NDLEV)
      INTEGER             ICNTE,       IL,          ILA(IL),     ISA(IL)
      INTEGER             IZ,          IZ0,         NDLEV,       NDMET
      INTEGER             NPL
      LOGICAL             LSS04A(NDLEV,NDMET)
      REAL*8              XJA(IL)
```

3.52 b8spln: Subroutine b8spln from library adas2xx

```
C
      SUBROUTINE B8SPLN( NTDIM , NDDIM ,
     &                   ITA    , IDA    , ITVAL   , IDVAL  ,
     &                   TETA   , TEDA   , TOUT    , DOUT   ,
     &                   CINA   ,          COUTA   ,
     &                                     LTRNG   , LDRNG
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: B8SPLN ******************  **
C
C  PURPOSE:
C          PERFORMS CUBIC SPLINE ON LOG(TEMPERATURE AND DENSITY)
C          VERSUS LOG(COLLISIONAL-RADIATIVE MATRIX COEFFICIENTS)
C          INPUT DATA
C
C          USING  TWO-WAY SPLINES IT CALCULATES  THE  INTERPOL. COEFFTS.
C          FOR  'ITVAL' ELECTRON TEMPERATURES  AND  'IDVAL' DENSITIES
C          FROM THE TWO-DIMENSIONAL TABLE OF TEMPERATURES/DENSITIES READ
C          IN FROM THE INPUT FILE. IF A  VALUE  CANNOT  BE  INTERPOLATED
C          USING SPLINES IT IS EXTRAPOLATED VIA 'XXSPLE'.
C
C  CALLING PROGRAM: ADAS208/B8GETP
C
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  NTDIM   = MAX NUMBER OF ELECTRON TEMPERATURES ALLOWED
C  INPUT : (I*4)  NDDIM   = MAX NUMBER OF ELECTRON DENSITIES    ALLOWED
C
C  INPUT : (I*4)  ITA     = INPUT DATA  : NUMBER OF ELECTRON TEMPERA-
C                           TURES
C  INPUT : (I*4)  IDA     = INPUT DATA  : NUMBER OF ELECTRON DENSIT-
C                           IES
C  INPUT : (I*4)  ITVAL   = OUTPUT DATA : NUMBER OF TEMPERATURES
C  INPUT : (I*4)  IDVAL   = OUTPUT DATA : NUMBER OF DENSITIES
C
C  INPUT : (R*8)  TETA()  = INPUT DATA  : ELECTRON TEMPERATURES (K)
C  INPUT : (R*8)  TEDA()  = INPUT DATA  : ELECTRON DENSITIES (CM-3)
C  INPUT : (R*8)  TOUT()  = OUTPUT DATA : ELECTRON TEMPERATURES (K)
C  INPUT : (R*8)  DOUT()  = OUTPUT DATA : ELECTRON DENSITIES (CM-3)
C
C
C  INPUT : (R*8)  CINA(,)  =INPUT DATA FILE: FULL SET OF COLL. RAD.
C                           COEFFICIENTS FOR THE DATA-BLOCK BEING
C                           ANALYSED.
C                           1ST DIMENSION: ELECTRON TEMPERATURE INDEX
C                           2ND DIMENSION: ELECTRON DENSITY     INDEX
C  OUTPUT: (R*8)  COUTA(,)= SPLINE INTERPOLATED COLL. RAD. CEOFFICIENTS
C                           THE USER ENTERED TEMPERATURES AND DENSITIES
C                           1ST DIMENSION: ELECTRON TEMPERATURE INDEX
C                           2ND DIMENSION: ELECTRON DENSITY     INDEX
C
C  OUTPUT: (L*4)  LTRNG()=  .TRUE.  => OUTPUT 'COUTA()' VALUE WAS INTER-
C                                      POLATED  FOR  THE  USER  ENTERED
C                                      ELECTRON TEMPERATURE 'TOUT'()'.
C                           .FALSE. => OUTPUT 'COUTA()' VALUE WAS EXTRA-
C                                      POLATED  FOR  THE  USER  ENTERED
C                                      ELECTRON TEMPERATURE 'TOUT'()'.
```

```
C                           DIMENSION: TEMPERATURE/DENSITY PAIR INDEX
C
C  OUTPUT: (L*4)  LDRNG()=  .TRUE.  => OUTPUT 'COUTA()' VALUE WAS INTER-
C                                      POLATED  FOR  THE  USER  ENTERED
C                                      ELECTRON DENSITY 'DOUT()'.
C                           .FALSE. => OUTPUT 'COUTA()' VALUE WAS EXTRA-
C                                      POLATED  FOR  THE  USER  ENTERED
C                                      ELECTRON DENSITY 'DOUT()'.
C                           DIMENSION: TEMPERATURE/DENSITY PAIR INDEX
C
C          (I*4)  NIN     = PARAMETER = MAX. NO. OF INPUT  TEMP/DENSITY
C                                       VALUES. MUST BE >= 'ITA'&'IDA'
C          (I*4)  NOUT    = PARAMETER = MAX. NO. OF OUTPUT TEMP/DENSITY
C                                       PAIRS.  MUST BE >= 'ITVAL'
C          (I*4)  IED     = ARRAY SUBSCRIPT USED INPUT  FILE  ELECTRON
C                           DENSITIES.
C          (I*4)  IET     = ARRAY SUBSCRIPT USED INPUT  FILE  ELECTRON
C                           TEMPERATURES.
C          (I*4)  IT      = ARRAY  SUBSCRIPT  USED  FOR  USER  ENTERED
C                           TEMPERATURES.
C          (I*4)  IN      = ARRAY  SUBSCRIPT  USED  FOR  USER  ENTERED
C                           DENSITIES.
C          (I*4)  IOPT    = DEFINES THE BOUNDARY DERIVATIVES FOR THE
C                           SPLINE ROUTOUT'E 'XXSPLE', SEE 'XXSPLE'.
C                           (VALID VALUES = <0, 0, 1, 2, 3, 4)
C
C          (L*4)  LSETX   = .TRUE.  => SET UP SPLINE PARAMETERS RELATOUT'G
C                                      TO 'XIN' AXIS.
C                           .FALSE. => DO NOT SET UP SPLINE PARAMETERS
C                                      RELATOUT'G TO 'XIN' AXIS.
C                                      (I.E. THEY WERE SET IN A PREVIOUS
C                                            CALL )
C                           (VALUE SET TO .FALSE. BY 'XXSPLE')
C
C          (R*8)  R8FUN1  = FUNCTION - (SEE ROUTOUT'ES SECTION BELOW)
C
C          (R*8)  XIN()   = 1) LOG( DATA FILE ELECTRON DENSITIES    )
C                           2) LOG( DATA FILE ELECTRON TEMPERATURES )
C          (R*8)  YIN()   = LOG( INPUT COLL. RAD COEFFTS.)
C          (R*8)  XOUT()  = 1) LOG( SCALED USER ENTERED ELECTRON DENS. )
C                           2) LOG( SCALED USER ENTERED ELECTRON TEMPS.)
C          (R*8)  YOUT()  = LOG( OUTPUT GENERATED IONIZATIONS/PHOTON )
C          (R*8)  YPASS(,)= LOG( COL. RAD. COEFFTS.) INTERMEDIATE ARRAY
C                           WHICH   STORES   INTERPOLATED/EXTRAPOLATED
C                           VALUES  BETWEEN  THE  TWO  SPLINE SECTIONS.
C                           SECTIONS.
C          (R*8)  DF()    = SPLINE INTERPOLATED DERIVATIVES
C
C
C NOTE:
C
C ROUTOUT'ES:
C          ROUTOUT'E  SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          XXSPLE     ADAS      SPLINE SUBROUTOUT'E (EXTENDED DIAGNOSTICS)
C          R8FUN1     ADAS      REAL*8 FUNCTION: ( X -> X )
C
C AUTHOR:  H.P. SUMMERS
C          K1/1/57
C          JET EXT. 4941
C
```

```
C DATE:    15/07/92
C
C
C*====================================================================*
C UNIX-IDL PORT:
C
C AUTHOR: DAVID H BROOKS, UNIVERSITY OF STRATHCLYDE
C
C DATE: UNKNOWN
C
C*====================================================================*
C PUT UNDER SCCS CONTROL:
C
C VERSION: 1.1 DATE: 10/05/96
C MODIFIED: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C     - FIRST PUT UNDER SCCS
C
C VERSION: 1.2 DATE: 23/05/96
C MODIFIED: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C     - INCREASED NOUT TO 35
C
C VERSION: 1.3 DATE: 30/09/96
C MODIFIED: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C     - INCREASED NIN TO 35
C
C--------------------------------------------------------------------
C
C--------------------------------------------------------------------
      INTEGER             IDA,         IDVAL,       ITA,         ITVAL
      INTEGER             NDDIM,       NTDIM
      LOGICAL             LDRNG(IDVAL),             LTRNG(ITVAL)
      REAL*8              CINA(NTDIM,NDDIM),        COUTA(NTDIM,NDDIM)
      REAL*8              DOUT(IDVAL), TEDA(IDA),   TETA(ITA)
      REAL*8              TOUT(ITVAL)
```

3.53 b8splt: Subroutine b8splt from library adas2xx

```
C
      SUBROUTINE B8SPLT( NTDIM , NDDIM ,
     &                   ITA    , IDA    , ITVAL   , IDVAL  ,
     &                   TETA   , TEDA   , TOUT    , DOUT   ,
     &                   CINA   ,          COUTA   ,
     &                                     LTRNG   , LDRNG
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: B8SPLT *********************
C
C  PURPOSE:
C          PERFORMS CUBIC SPLINE ON LOG(TEMPERATURE AND DENSITY)
C          VERSUS LOG(COLLISIONAL-RADIATIVE MATRIX COEFFICIENTS)
C          INPUT DATA
C
C          USING  TWO-WAY SPLINES IT CALCULATES  THE  INTERPOL. COEFFTS.
C          FOR  'ITVAL' ELECTRON TEMPERATURES  AND  'IDVAL' DENSITIES
C          FROM THE TWO-DIMENSIONAL TABLE OF TEMPERATURES/DENSITIES READ
C          IN FROM THE INPUT FILE. IF A  VALUE  CANNOT  BE  INTERPOLATED
C          USING SPLINES IT IS EXTRAPOLATED VIA 'XXSPLE'.
C
C          This is a special version of B8SPLN which examines the input
C          data for zeros (effective zeros of 1.0D-72) and eliminates
C          these from the spline. This is to avoid NaNQ propagating
C          throughout the population calculation. So far only one
C          array is affected - PCRMAT in B8GETP. Only do a spline
C          fit if at least half of the points are present.
C
C  CALLING PROGRAM: ADAS208/B8GETP
C
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  NTDIM   = MAX NUMBER OF ELECTRON TEMPERATURES ALLOWED
C  INPUT : (I*4)  NDDIM   = MAX NUMBER OF ELECTRON DENSITIES    ALLOWED
C
C  INPUT : (I*4)  ITA     = INPUT DATA  : NUMBER OF ELECTRON TEMPERA-
C                           TURES
C  INPUT : (I*4)  IDA     = INPUT DATA  : NUMBER OF ELECTRON DENSIT-
C                           IES
C  INPUT : (I*4)  ITVAL   = OUTPUT DATA : NUMBER OF TEMPERATURES
C  INPUT : (I*4)  IDVAL   = OUTPUT DATA : NUMBER OF DENSITIES
C
C  INPUT : (R*8)  TETA()  = INPUT DATA  : ELECTRON TEMPERATURES (K)
C  INPUT : (R*8)  TEDA()  = INPUT DATA  : ELECTRON DENSITIES (CM-3)
C  INPUT : (R*8)  TOUT()  = OUTPUT DATA : ELECTRON TEMPERATURES (K)
C  INPUT : (R*8)  DOUT()  = OUTPUT DATA : ELECTRON DENSITIES (CM-3)
C
C
C  INPUT : (R*8)  CINA(,)  =INPUT DATA FILE: FULL SET OF COLL. RAD.
C                           COEFFICIENTS FOR THE DATA-BLOCK BEING
C                           ANALYSED.
C                           1ST DIMENSION: ELECTRON TEMPERATURE INDEX
C                           2ND DIMENSION: ELECTRON DENSITY     INDEX
C  OUTPUT: (R*8)  COUTA(,)= SPLINE INTERPOLATED COLL. RAD. CEOFFICIENTS
C                           THE USER ENTERED TEMPERATURES AND DENSITIES
C                           1ST DIMENSION: ELECTRON TEMPERATURE INDEX
C                           2ND DIMENSION: ELECTRON DENSITY     INDEX
```

```
C
C  OUTPUT: (L*4)  LTRNG()=  .TRUE.  => OUTPUT 'COUTA()' VALUE WAS INTER-
C                                     POLATED  FOR  THE  USER  ENTERED
C                                     ELECTRON TEMPERATURE 'TOUT'()'.
C                           .FALSE. => OUTPUT 'COUTA()' VALUE WAS EXTRA-
C                                     POLATED  FOR  THE  USER  ENTERED
C                                     ELECTRON TEMPERATURE 'TOUT'()'.
C                           DIMENSION: TEMPERATURE/DENSITY PAIR INDEX
C
C  OUTPUT: (L*4)  LDRNG()=  .TRUE.  => OUTPUT 'COUTA()' VALUE WAS INTER-
C                                     POLATED  FOR  THE  USER  ENTERED
C                                     ELECTRON DENSITY 'DOUT()'.
C                           .FALSE. => OUTPUT 'COUTA()' VALUE WAS EXTRA-
C                                     POLATED  FOR  THE  USER  ENTERED
C                                     ELECTRON DENSITY 'DOUT()'.
C                           DIMENSION: TEMPERATURE/DENSITY PAIR INDEX
C
C          (I*4)  NIN     = PARAMETER = MAX. NO. OF INPUT  TEMP/DENSITY
C                                       VALUES. MUST BE >= 'ITA'&'IDA'
C          (I*4)  NOUT    = PARAMETER = MAX. NO. OF OUTPUT TEMP/DENSITY
C                                       PAIRS.  MUST BE >= 'ITVAL'
C          (I*4)  IED     = ARRAY SUBSCRIPT USED INPUT  FILE  ELECTRON
C                           DENSITIES.
C          (I*4)  IET     = ARRAY SUBSCRIPT USED INPUT  FILE  ELECTRON
C                           TEMPERATURES.
C          (I*4)  IT      = ARRAY  SUBSCRIPT  USED  FOR  USER  ENTERED
C                           TEMPERATURES.
C          (I*4)  IN      = ARRAY  SUBSCRIPT  USED  FOR  USER  ENTERED
C                           DENSITIES.
C          (I*4)  IOPT    = DEFINES THE BOUNDARY DERIVATIVES FOR THE
C                           SPLINE ROUTOUT'E 'XXSPLE', SEE 'XXSPLE'.
C                           (VALID VALUES = <0, 0, 1, 2, 3, 4)
C
C          (L*4)  LSETX   = .TRUE.  => SET UP SPLINE PARAMETERS RELATOUT'G
C                                      TO 'XIN' AXIS.
C                           .FALSE. => DO NOT SET UP SPLINE PARAMETERS
C                                      RELATOUT'G TO 'XIN' AXIS.
C                                      (I.E. THEY WERE SET IN A PREVIOUS
C                                            CALL )
C                           (VALUE SET TO .FALSE. BY 'XXSPLE')
C
C          (R*8)  R8FUN1  = FUNCTION - (SEE ROUTOUT'ES SECTION BELOW)
C
C          (R*8)  XIN()   = 1) LOG( DATA FILE ELECTRON DENSITIES    )
C                           2) LOG( DATA FILE ELECTRON TEMPERATURES )
C          (R*8)  YIN()   = LOG( INPUT COLL. RAD COEFFTS.)
C          (R*8)  XOUT()  = 1) LOG( SCALED USER ENTERED ELECTRON DENS. )
C                           2) LOG( SCALED USER ENTERED ELECTRON TEMPS.)
C          (R*8)  YOUT()  = LOG( OUTPUT GENERATED IONIZATIONS/PHOTON )
C          (R*8)  YPASS(,)= LOG( COL. RAD. COEFFTS.) INTERMEDIATE ARRAY
C                           WHICH   STORES   INTERPOLATED/EXTRAPOLATED
C                           VALUES  BETWEEN  THE  TWO  SPLINE SECTIONS.
C                           SECTIONS.
C          (R*8)  DF()    = SPLINE INTERPOLATED DERIVATIVES
C
C
C NOTE:
C
C ROUTOUT'ES:
C          ROUTOUT'E  SOURCE    BRIEF DESCRIPTION
C          ---------------------------------------------------------
```

```
C          XXSPLE     ADAS      SPLINE SUBROUTOUT'E (EXTENDED DIAGNOSTICS)
C          R8FUN1     ADAS      REAL*8 FUNCTION: ( X -> X )
C
C AUTHOR:  H.P. SUMMERS
C          K1/1/57
C          JET EXT. 4941
C
C DATE:    15/07/92
C
C
C
C VERSION : 1.1 DATE: 10/05/96
C MODIFIED: Martin O'Mullane
C     - First version
C
C VERSION : 1.2 DATE: 24/09/2004
C MODIFIED: Martin O'Mullane
C     - The check to avoid integrating over zeros in the input
C             can result in no valid points. This causes xxsple an
C             out of bounds error in xxsple. Add a check to avoid
C             the call in this case.
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IDA,          IDVAL,       ITA,          ITVAL
      INTEGER             NDDIM,        NTDIM
      LOGICAL             LDRNG(IDVAL),              LTRNG(ITVAL)
      REAL*8              CINA(NTDIM,NDDIM),         COUTA(NTDIM,NDDIM)
      REAL*8              DOUT(IDVAL), TEDA(IDA),    TETA(ITA)
      REAL*8              TOUT(ITVAL)
```

3.54 b8stkb: Subroutine b8stkb from library adas2xx

```
C
      SUBROUTINE B8STKB( NDTEM  , NDLEV  , NDMET  ,
     &                   IT     , NORD   ,
     &                            IORDR  ,
     &                   CMAT   , VEC    ,
     &                   IP     ,
     &                   STV
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: B8STKB *********************
C
C  PURPOSE: TO STACK UP IN  'STV'  THE  RECOMBINATION CONTRIBUTION  FOR
C           EACH  NON-METASTABLE/ORDINARY  EXCITED  LEVEL  FOR  A GIVEN
C           TEMPERATURE AND DENSITY.
C
C  CALLING PROGRAM:  ADAS205/ADAS206
C
C  SUBROUTINE:
C
C  INPUT :  (I*4)  NDTEM   = MAXIMUM NUMBER OF TEMPERATURES ALLOWED
C  INPUT :  (I*4)  NDLEV   = MAXIMUM NUMBER OF ENERGY LEVELS ALLOWED
C  INPUT :  (I*4)  NDMET   = MAXIMUM NUMBER OF METASTABLES ALLOWED
C
C  INPUT :  (I*4)  IT      = INDEX DENOTING THE TEMPERATURE
C  INPUT :  (I*4)  NORD    = NUMBER OF ORDINARY EXCITED LEVELS
C
C  INPUT :  (I*4)  IORDR() =INDEX OF ORDINARY EXCITED LEVELS IN COMPLETE
C                           LEVEL LIST.
C                           (ARRAY SIZE = 'NDLEV' )
C
C  INPUT :  (R*8)  CMAT(,) = INVERTED   RATE   MATRIX   COVERING   ALL
C                           NON-METASTABLE/ORDINARY EXCITED LEVELS
C                           TRANSITIONS.
C                           (UNITS: SEC)
C                           VALUES FOR GIVEN TEMPERATURE AND DENSITY.
C                           1st DIMENSION: ORDINARY EXCITED LEVEL INDEX
C                           2nd DIMENSION: ORDINARY EXCITED LEVEL INDEX
C  INPUT :  (I*4)  IP      = PARENT INDEX
C
C  INPUT :  (R*8)  VEC(,,) = RECOMBINATION RATE COEFFT. VALUES.
C                           (UNITS: CM**3/SEC-1)
C                           VALUES FOR GIVEN TEMPERATURE AND DENSITY.
C                           1st DIMENSION: TEMPERATURE INDEX ('IT')
C                           2nd DIMENSION: CAPTURING LEVEL INDEX
C                           3rd DIMENSION: PARENT INDEX
C
C  OUTPUT:  (R*4)  STV()   = RECOMBINATION  CONTRIBUTION  FOR  EACH
C                           NON-METASTABLE/ORDINARY EXCITED LEVELS.
C                           (UNITS: CM**3)
C                           VALUES FOR GIVEN TEMPERATURE AND DENSITY.
C                           DIMENSION: ORDINARY EXCITED LEVEL INDEX
C
C           (I*4)  IS1     = ORDINARY EXCITED LEVEL INDEX
C           (I*4)  IS2     = ORDINARY EXCITED LEVEL INDEX
C
C           (R*8)  COEF    = VARIABLE USED TO SUM COEFFICIENT VALUES
C
C
```

```
C ROUTINES: NONE
C
C NOTE:
C        IF:     n  =  number of ordinary/non-metastable levels
C            R(nxn) = Rate matrix (SEC-1) covering transistions between
C                     all possible pairs of ordinary levels.
C                     row   : final   level
C                     column: initial level
C                     (Inverse R-1(nxn) = 'CMAT(,)' )
C            V(n)   = Recombination rate vector (CM**3 SEC-1) covering
C                     all ordinary levels.
C                     ( = 'VEC()' - ordinary level part ).
C            S(n)   = Recombination contribution vector (CM**3) covering
C                     all ordinary levels ( = 'STV()' ).
C
C           Therefore:  R(nxn).S(n) = V(n)
C
C            =>         S(n)  = R-1(nxn).V(n)
C
C
C
C AUTHOR:  HP SUMMERS  (UPGRADE OF BXSTKB BT PE BRIDEN)
C          K1/1/57
C          JET EXT. 4941
C
C DATE:    11/06/92
C
C UPDATE:  12/07/93  HPS - CHASNGE STV DIMENSION TO R*4
C*********************************************************************
C UNIX-IDL PORT:
C
C AUTHOR: DAVID H BROOKS, UNIVERSITY OF STRATHCLYDE
C
C DATE: UNKNOWN
C
C*********************************************************************
C PUT UNDER SCCS CONTROL:
C
C VERSION: 1.1 DATE: 10/05/96
C MODIFIED: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C     - FIRST PUT UNDER SCCS
C
C---------------------------------------------------------------------
C
C---------------------------------------------------------------------
      INTEGER             IORDR(NDLEV),             IP,          IT
      INTEGER             NDLEV,       NDMET,       NDTEM,       NORD
      REAL*8              CMAT(NDLEV,NDLEV)
      REAL                STV(NDLEV)
      REAL*8              VEC(NDTEM,NDLEV,NDMET)
```

3.55 b8stkd: Subroutine b8stkd from library adas2xx

```
C
      SUBROUTINE B8STKD( NDTEM  , NDLEV  , NDMET  ,
     &                   IT     , NORD   , NMET   ,
     &                            IORDR  , IMETR  ,
     &                   CC     , STV    ,
     &                   VEC    , IP     ,
     &                   VRED
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: B8STKD *********************
C
C  PURPOSE: TO STACK UP IN  'VRED'  THE RECOMBINATION RATE CONTRIBUTIONS
C           FOR  EACH  METASTABLE  LEVEL  FOR  A  GIVEN TEMPERATURE  AND
C           DENSITY.
C
C  CALLING PROGRAM:  ADAS205/ADAS206
C
C  SUBROUTINE:
C
C  INPUT :  (I*4)  NDTEM   = MAXIMUM NUMBER OF TEMPERATURES ALLOWED
C  INPUT :  (I*4)  NDLEV   = MAXIMUM NUMBER OF ENERGY LEVELS ALLOWED
C  INPUT :  (I*4)  NDMET   = MAXIMUM NUMBER OF METASTABLE LEVELS ALLOWED
C
C  INPUT :  (I*4)  IT      = INDEX DENOTING THE TEMPERATURE
C  INPUT :  (I*4)  IP      = PARENT INDEX
C  INPUT :  (I*4)  NORD    = NUMBER OF ORDINARY EXCITED LEVELS
C  INPUT :  (I*4)  NMET    = NUMBER OF METASTABLE LEVELS
C
C  INPUT :  (I*4)  IMETR() = INDEX OF METASTABLE IN COMPLETE LEVEL LIST
C                            (ARRAY SIZE = 'NDMET' )
C  INPUT :  (I*4)  IORDR() =INDEX OF ORDINARY EXCITED LEVELS IN COMPLETE
C                           LEVEL LIST.
C                           (ARRAY SIZE = 'NDLEV' )
C
C  INPUT :  (R*8)  CC(,)   = RATE MATRIX COVERING ALL TRANSITIONS
C                           (UNITS: SEC-1)
C                           VALUES FOR GIVEN TEMPERATURE AND DENSITY.
C                           1st DIMENSION: ENERGY LEVEL INDEX
C                           2nd DIMENSION: ENERGY LEVEL INDEX
C  INPUT :  (R*4)  STV()   = RECOMBINATION  CONTRIBUTION  FOR  EACH
C                           NON-METASTABLE/ORDINARY EXCITED LEVELS.
C                           (UNITS: CM**3)
C                           VALUES FOR GIVEN TEMPERATURE AND DENSITY.
C                           DIMENSION: ORDINARY EXCITED LEVEL INDEX
C  INPUT :  (R*8)  VEC(,,) = RECOMBINATION RATE COEFFT. VALUES.
C                           (UNITS: CM**3/SEC-1)
C                           VALUES FOR GIVEN TEMPERATURE AND DENSITY.
C                           1st DIMENSION: TEMPERATURE INDEX ('IT')
C                           2nd DIMENSION: CAPTURING LEVEL INDEX
C                           3ND DIMENSION: PARENT INDEX
C
C  OUTPUT:  (R*8)  VRED(,) = VECTOR  OF RECOMBINATION RATE CONTRIBUTIONS
C                           FOR EACH METASTABLE LEVEL.
C                           (UNITS: SEC-1)
C                           VALUES FOR GIVEN TEMPERATURE AND DENSITY.
C                           1ST DIMENSION: METASTABLE LEVEL INDEX
C                           2ND DIMENSION: PARENT INDEX
C
```

```
C           (I*4)  IM      = METASTABLE LEVEL ARRAY INDEX
C           (I*4)  IS      = ORDINARY EXCITED LEVEL INDEX
C
C
C ROUTINES: NONE
C
C NOTE:
C           VRED(IM,IP)   =    ( THE RECOMBINATION RATE FOR IM )
C                                          +
C                            SUM( (the transistion rate from ordinary
C                                  level IS to IM)  x  (the recombin-
C                                  ation  contribution  for  ordinary
C                                  level IS) )
C
C                            ABOVE SUM IS OVER ALL ORDINARY LEVELS.
C
C
C AUTHOR:  HP SUMMERS  (UPGRADE OF BXSTKD BY PE BRIDEN)
C          K1/1/57
C          JET EXT. 4941
C
C DATE:    11/06/92
C
C UPDATE:  12/07/93  HPS - CHANGE STV DIMENSION TO R*4
C**********************************************************************
C UNIX-IDL PORT:
C
C AUTHOR: DAVID H BROOKS, UNIVERSITY OF STRATHCLYDE
C
C DATE: UNKNOWN
C
C**********************************************************************
C PUT UNDER SCCS CONTROL:
C
C VERSION: 1.1 DATE: 10/05/96
C MODIFIED: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C     - FIRST PUT UNDER SCCS
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IMETR(NDMET),             IORDR(NDLEV)
      INTEGER             IP,          IT,          NDLEV,       NDMET
      INTEGER             NDTEM,       NMET,        NORD
      REAL*8              CC(NDLEV,NDLEV)
      REAL                STV(NDLEV)
      REAL*8              VEC(NDTEM,NDLEV,NDMET),   VRED(NDMET,NDMET)
```

3.56 b8stke: Subroutine b8stke from library adas2xx

```
C
      SUBROUTINE B8STKE( NDTEM  , NDLEV  , NDMET  ,
     &                   IT     , NORD   ,
     &                            IORDR  ,
     &                   DENS   ,
     &                   CMAT   , VEC    , V3     ,
     &                   IP     ,
     &                   STV
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: B8STKE *********************
C
C  PURPOSE: TO STACK UP IN  'STV'  THE  RECOMBINATION CONTRIBUTION  FOR
C           EACH  NON-METASTABLE/ORDINARY  EXCITED  LEVEL  FOR  A GIVEN
C           TEMPERATURE AND DENSITY, BUT ADDING A THREE-BODY
C           RECOMBINATION PART FROM V3
C
C  CALLING PROGRAM:  ADAS208
C
C  SUBROUTINE:
C
C  INPUT :  (I*4)  NDTEM   = MAXIMUM NUMBER OF TEMPERATURES ALLOWED
C  INPUT :  (I*4)  NDLEV   = MAXIMUM NUMBER OF ENERGY LEVELS ALLOWED
C  INPUT :  (I*4)  NDMET   = MAXIMUM NUMBER OF METASTABLES ALLOWED
C
C  INPUT :  (I*4)  IT      = INDEX DENOTING THE TEMPERATURE
C  INPUT :  (I*4)  NORD    = NUMBER OF ORDINARY EXCITED LEVELS
C
C  INPUT :  (I*4)  IORDR() =INDEX OF ORDINARY EXCITED LEVELS IN COMPLETE
C                           LEVEL LIST.
C                           (ARRAY SIZE = 'NDLEV' )
C
C  INPUT :  (R*8)  DENS    = ELECTRON DENSITY (CM-3)
C  INPUT :  (R*8)  CMAT(,) = INVERTED   RATE   MATRIX   COVERING   ALL
C                           NON-METASTABLE/ORDINARY EXCITED LEVELS
C                           TRANSITIONS.
C                           (UNITS: SEC)
C                           VALUES FOR GIVEN TEMPERATURE AND DENSITY.
C                           1st DIMENSION: ORDINARY EXCITED LEVEL INDEX
C                           2nd DIMENSION: ORDINARY EXCITED LEVEL INDEX
C  INPUT :  (I*4)  IP      = PARENT INDEX
C
C  INPUT :  (R*8)  VEC(,,) = RECOMBINATION RATE COEFFT. VALUES.
C                           (UNITS: CM**3/SEC-1)
C                           VALUES FOR GIVEN TEMPERATURE AND DENSITY.
C                           1st DIMENSION: TEMPERATURE INDEX ('IT')
C                           2nd DIMENSION: CAPTURING LEVEL INDEX
C                           3rd DIMENSION: PARENT INDEX
C  INPUT :  (R*8)  V3(,)   = THREE-BODY RECOMB. RATE COEFFT. VALUES.
C                           (UNITS: CM**6/SEC-1)
C                           VALUES FOR A SPECIFIED TEMPERATURE.
C                           1ST DIMENSION: CAPTURING LEVEL INDEX
C                           2ND DIMENSION: PARENT INDEX
C
C  OUTPUT:  (R*4)  STV()   = RECOMBINATION  CONTRIBUTION  FOR  EACH
C                           NON-METASTABLE/ORDINARY EXCITED LEVELS.
C                           (UNITS: CM**3)
C                           VALUES FOR GIVEN TEMPERATURE AND DENSITY.
```

```
C                            DIMENSION: ORDINARY EXCITED LEVEL INDEX
C
C           (I*4)  IS1     = ORDINARY EXCITED LEVEL INDEX
C           (I*4)  IS2     = ORDINARY EXCITED LEVEL INDEX
C
C           (R*8)  COEF    = VARIABLE USED TO SUM COEFFICIENT VALUES
C
C
C ROUTINES: NONE
C
C NOTE:
C        IF:     n  =  number of ordinary/non-metastable levels
C            R(nxn) = Rate matrix (SEC-1) covering transistions between
C                     all possible pairs of ordinary levels.
C                     row   : final   level
C                     column: initial level
C                     (Inverse R-1(nxn) = 'CMAT(,)' )
C            V(n)   = Recombination rate vector (CM**3 SEC-1) covering
C                     all ordinary levels.
C                     ( = 'VEC()' - ordinary level part ).
C            S(n)   = Recombination contribution vector (CM**3) covering
C                     all ordinary levels ( = 'STV()' ).
C
C           Therefore:  R(nxn).S(n) = V(n)
C
C            =>         S(n)  = R-1(nxn).V(n)
C
C
C
C AUTHOR:  HP SUMMERS  (UPGRADE OF BXSTKB BT PE BRIDEN)
C          K1/1/57
C          JET EXT. 4941
C
C DATE:    11/06/92
C
C UPDATE:  12/07/93  HPS - CHANGE STV DIMENSION TO R*4
C**********************************************************************
C UNIX-IDL PORT:
C
C AUTHOR: DAVID H BROOKS, UNIVERSITY OF STRATHCLYDE
C
C DATE: UNKNOWN
C
C**********************************************************************
C PUT UNDER SCCS CONTROL:
C
C VERSION: 1.1 DATE: 10/05/96
C MODIFIED: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C     - FIRST PUT UNDER SCCS
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IORDR(NDLEV),             IP,          IT
      INTEGER             NDLEV,       NDMET,       NDTEM,       NORD
      REAL*8              CMAT(NDLEV,NDLEV),        DENS
      REAL                STV(NDLEV)
      REAL*8              V3(NDLEV,NDMET)
      REAL*8              VEC(NDTEM,NDLEV,NDMET)
```

3.57 b8stkf: Subroutine b8stkf from library adas2xx

```
C
      SUBROUTINE B8STKF( NDTEM  , NDLEV  , NDMET  ,
     &                   IT     , NORD   , NMET   ,
     &                            IORDR  , IMETR  ,
     &                   CC     , STV    ,
     &                   DENS   ,
     &                   VEC    , V3     , IP     ,
     &                   VRED
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: B8STKF *********************
C
C  PURPOSE: TO STACK UP IN  'VRED'  THE RECOMBINATION RATE CONTRIBUTIONS
C           FOR  EACH  METASTABLE  LEVEL  FOR  A  GIVEN TEMPERATURE  AND
C           DENSITY AND INCLUDE THREE-BODY RECOMBINATION PART
C
C  CALLING PROGRAM:  ADAS208
C
C  SUBROUTINE:
C
C  INPUT :  (I*4)  NDTEM   = MAXIMUM NUMBER OF TEMPERATURES ALLOWED
C  INPUT :  (I*4)  NDLEV   = MAXIMUM NUMBER OF ENERGY LEVELS ALLOWED
C  INPUT :  (I*4)  NDMET   = MAXIMUM NUMBER OF METASTABLE LEVELS ALLOWED
C
C  INPUT :  (I*4)  IT      = INDEX DENOTING THE TEMPERATURE
C  INPUT :  (I*4)  IP      = PARENT INDEX
C  INPUT :  (I*4)  NORD    = NUMBER OF ORDINARY EXCITED LEVELS
C  INPUT :  (I*4)  NMET    = NUMBER OF METASTABLE LEVELS
C
C  INPUT :  (I*4)  IMETR() = INDEX OF METASTABLE IN COMPLETE LEVEL LIST
C                            (ARRAY SIZE = 'NDMET' )
C  INPUT :  (I*4)  IORDR() =INDEX OF ORDINARY EXCITED LEVELS IN COMPLETE
C                           LEVEL LIST.
C                           (ARRAY SIZE = 'NDLEV' )
C
C  INPUT :  (R*8)  DENS    = ELECTRON DENSITY (CM-3)
C  INPUT :  (R*8)  CC(,)   = RATE MATRIX COVERING ALL TRANSITIONS
C                           (UNITS: SEC-1)
C                           VALUES FOR GIVEN TEMPERATURE AND DENSITY.
C                           1st DIMENSION: ENERGY LEVEL INDEX
C                           2nd DIMENSION: ENERGY LEVEL INDEX
C  INPUT :  (R*4)  STV()   = RECOMBINATION  CONTRIBUTION  FOR  EACH
C                           NON-METASTABLE/ORDINARY EXCITED LEVELS.
C                           (UNITS: CM**3)
C                           VALUES FOR GIVEN TEMPERATURE AND DENSITY.
C                           DIMENSION: ORDINARY EXCITED LEVEL INDEX
C  INPUT :  (R*8)  VEC(,,) = RECOMBINATION RATE COEFFT. VALUES.
C                           (UNITS: CM**3/SEC-1)
C                           VALUES FOR GIVEN TEMPERATURE AND DENSITY.
C                           1st DIMENSION: TEMPERATURE INDEX ('IT')
C                           2nd DIMENSION: CAPTURING LEVEL INDEX
C                           3ND DIMENSION: PARENT INDEX
C  INPUT :  (R*8)  V3(,)   = THREE-BODY RECOMB. RATE COEFFT. VALUES.
C                           (UNITS: CM**6/SEC-1)
C                           VALUES FOR A SPECIFIC TEMPERATURE.
C                           1ST DIMENSION: CAPTURING LEVEL INDEX
C                           2ND DIMENSION: PARENT INDEX
C
```

```
C  OUTPUT:  (R*8)  VRED(,) = VECTOR  OF RECOMBINATION RATE CONTRIBUTIONS
C                            FOR EACH METASTABLE LEVEL.
C                            (UNITS: SEC-1)
C                            VALUES FOR GIVEN TEMPERATURE AND DENSITY.
C                            1ST DIMENSION: METASTABLE LEVEL INDEX
C                            2ND DIMENSION: PARENT INDEX
C
C           (I*4)  IM      = METASTABLE LEVEL ARRAY INDEX
C           (I*4)  IS      = ORDINARY EXCITED LEVEL INDEX
C
C
C ROUTINES: NONE
C
C NOTE:
C           VRED(IM,IP)   =    ( THE RECOMBINATION RATE FOR IM )
C                                              +
C                            SUM( (the transistion rate from ordinary
C                                  level IS to IM)  x  (the recombin-
C                                  ation  contribution  for  ordinary
C                                  level IS) )
C
C                            ABOVE SUM IS OVER ALL ORDINARY LEVELS.
C
C
C AUTHOR:  HP SUMMERS  (UPGRADE OF BXSTKD BY PE BRIDEN)
C          K1/1/57
C          JET EXT. 4941
C
C DATE:    11/06/92
C
C UPDATE:  12/07/93  HPS - CHANGE STV DIMENSION TO R*4
C-----------------------------------------------------------------------
C UNIX-IDL PORT:
C
C AUTHOR: DAVID H BROOKS, UNIVERSITY OF STRATHCLYDE
C
C DATE: UNKNOWN
C
C-----------------------------------------------------------------------
C PUT UNDER SCCS CONTROL:
C
C VERSION: 1.1 DATE: 10/05/96
C MODIFIED: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C     - FIRST PUT UNDER SCCS
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IMETR(NDMET),             IORDR(NDLEV)
      INTEGER             IP,          IT,          NDLEV,       NDMET
      INTEGER             NDTEM,       NMET,        NORD
      REAL*8              CC(NDLEV,NDLEV),          DENS
      REAL                STV(NDLEV)
      REAL*8              V3(NDLEV,NDMET)
      REAL*8              VEC(NDTEM,NDLEV,NDMET),   VRED(NDMET,NDMET)
```

3.58 b8stvm: Subroutine b8stvm from library adas2xx

```
C
      SUBROUTINE B8STVM( NDMET  ,
     &                   NMET   ,
     &                   CRMAT  ,
     &                   IP     ,
     &                   VRED   ,
     &                   STVM
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: B8STVM *********************
C
C  PURPOSE: TO  CALCULATE  AND STACK UP IN  'STVM'  THE METASTABLE LEVEL
C           RECOMBINATION  COEFFICIENTS  FOR  A  GIVEN  TEMPERATURE  AND
C           DENSITY.
C
C  CALLING PROGRAM:  ADAS205/ADAS206
C
C  SUBROUTINE:
C
C  INPUT :  (I*4)  NDMET   = MAXIMUM NUMBER OF METASTABLE LEVELS ALLOWED
C
C  INPUT :  (I*4)  NMET    = NUMBER OF METASTABLE LEVELS
C
C  INPUT :  (R*8)  CRMAT(,)= INVERTED  METASTABLE  LEVEL   RATE   MATRIX
C                            COVERING ALL TRANSITIONS BETWEEN METASTABLE
C                            LEVELS EXCEPT THOSE INVOLVING LEVEL 1.
C                            (UNITS: SEC)
C                            VALUES FOR GIVEN TEMPERATURE AND DENSITY.
C                            1st DIMENSION: METASTABLE LEVEL INDEX - 1
C                            2nd DIMENSION: METASTABLE LEVEL INDEX - 1
C           (I*4)  IP       = PARENT INDEX
C
C  INPUT :  (R*8)  VRED(,) = VECTOR  OF RECOMBINATION RATE CONTRIBUTIONS
C                            FOR EACH METASTABLE LEVEL.
C                            (UNITS: SEC-1)
C                            VALUES FOR GIVEN TEMPERATURE AND DENSITY.
C                            DIMENSION: METASTABLE LEVEL INDEX
C
C  OUTPUT:  (R*8)  STVM()  = RECOMBINATION  CONTRIBUTION  FOR  EACH
C                            METASTABLE LEVEL. (UNITS: CM**3)
C                            VALUES FOR GIVEN TEMPERATURE AND DENSITY.
C                            (LEVEL 1 IS TAKEN AS ZERO)
C                            DIMENSION: METASTABLE LEVEL INDEX
C
C           (I*4)  IM1      = METASTABLE LEVEL ARRAY INDEX
C           (I*4)  IM2      = METASTABLE LEVEL ARRAY INDEX
C
C
C ROUTINES: NONE
C
C NOTE:
C           STVM(IM1)       SUM( (the transistion rate from IM2 to IM1)
C                                x (the recombination rate contribution
C                                   for metastable level IM2) )
C
C                           (IM1 & IM2 = METASTABLE LEVEL INDEX)
C
C                           ABOVE SUM IS OVER ALL METASTABLE LEVELS
```

```
C                           EXCEPT LEVEL ONE.
C
C
C AUTHOR:  HP SUMMERS ( UPGRADE OF BXSTVM BY PE BRIDEN)
C          K1/1/57
C          JET EXT. 4941
C
C DATE:    11/06/92
C
C*********************************************************************
C UNIX-IDL PORT:
C
C AUTHOR: DAVID H BROOKS, UNIVERSITY OF STRATHCLYDE
C
C DATE: UNKNOWN
C
C*********************************************************************
C PUT UNDER SCCS CONTROL:
C
C VERSION: 1.1 DATE: 10/05/96
C MODIFIED: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C     - FIRST PUT UNDER SCCS
C
C---------------------------------------------------------------------
C
C---------------------------------------------------------------------
      INTEGER             IP,          NDMET,       NMET
      REAL*8              CRMAT(NDMET,NDMET),       STVM(NDMET)
      REAL*8              VRED(NDMET,NDMET)
```

3.59 b8toth: Subroutine b8toth from library adas2xx

```
C
      SUBROUTINE B8TOTH( NDLEV   , NDMET  , NDTEM , NDDEN ,
     &                   NORD    , NMET   , NPL   ,
     &                   IORDR   , IMETR  ,
     &                   IT      , MAXT   , IN    , MAXD  ,
     &                   RATPIA  ,
     &                   STVHM   , STVH   ,
     &                   PLA1    ,
     &                   PHA     , PH
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: B8TOTH *********************
C
C  PURPOSE: TO CALCULATE TOTAL CHARGE EXCHANGE DRIVEN LINE POWER.
C
C  NOTE: CODE EXECUTES FOR ONE TEMPERATURE AND DENSITY INDEX AT A TIME
C
C  CALLING PROGRAM:  ADAS208
C
C  SUBROUTINE:
C
C  INPUT : (I*4) NDLEV    = PARAMETER = MAX. NO. OF LEVELS ALLOWED
C  INPUT : (I*4) NDMET    = PARAMETER = MAX. NO. OF METASTABLES ALLOWED
C  INPUT : (I*4) NDTEM    = PARAMETER = MAX. NO. OF TEMPERATURES ALLOWED
C  INPUT : (I*4) NDDEN    = PARAMETER = MAX. NO. OF DENSITIES ALLOWED
C
C  INPUT : (I*4) NORD     = NUMBER OF ORD. LEVELS (1 <= NORD <= 'NDLEV')
C  INPUT : (I*4) NMET     = NUMBER OF METASTABLES (1 <= NMET <= 'NDMET')
C  INPUT : (I*4) NPL      = NUMBER OF PARENT METASTABLES (NPL<= 'NDMET')
C
C  INPUT : (I*4) IORDR()  = INDEX OF ORDINARY LEVELS IN COMPLETE LEVEL
C                           LIST (ARRAY SIZE = 'NDLEV' )
C  INPUT : (I*4) IMETR()  = INDEX OF METASTABLE IN COMPLETE LEVEL LIST
C                           (ARRAY SIZE = 'NDMET' )
C  INPUT : (I*4) IT       = CURRENT TEMPERATURE INDEX
C  INPUT : (I*4) MAXT     = NUMBER OF INPUT TEMPERATURES ( 1 -> 'NDTEM')
C  INPUT : (I*4) IN       = CURRENT DENSITY INDEX
C  INPUT : (I*4) MAXD     = NUMBER OF INPUT DENSITIES ( 1 -> 'NDDEN')
C  INPUT : (R*8) RATPIA(,)= RATIO ( N(Z+1)/N(Z)  STAGE ABUNDANCIES )
C                            1ST DIMENSION: DENS INDEX
C                            2ND DIMENSION: PARENT INDEX
C
C  INPUT : (R*8) STVHM(,,,)= METASTABLE LEVEL:
C                           CHARGE-EXCHANGE RECOMBINATION POPUL. PART
C                           (UNITS* CM**3/SEC-1)
C                             1st DIMENSION: METASTABLE INDEX
C                             2nd DIMENSION: TEMPERATURE INDEX
C                             3rd DIMENSION: DENSITY INDEX
C                             4TH DIMENSION: PARENT INDEX
C  INPUT : (R*4) STVH(,,,)= ORDINARY EXCITED LEVEL:
C                             CHARGE-EXCHANGE RECOMBINATION POPUL. PART
C                             (UNITS* CM**3/SEC-1)
C                             1st DIMENSION: ORDINARY LEVEL INDEX
C                             2nd DIMENSION: TEMPERATURE INDEX
C                             3rd DIMENSION: DENSITY INDEX
C                             4TH DIMENSION: PARENT INDEX
C  INPUT : (R*8)  PLA1()  = DIRECT LINE POWER LOSS FOR EACH LEVEL.
C                           (UNITS: ERGS SEC-1) (DIMENSION: LEVEL INDEX)
```

```
C
C  OUTPUT: (R*8)  PH(,,)  = TOTAL CX LINE POWER FOR PARENT. THIS IS
C                           THE SUM OF ALL EMISSIONS ORGINATING IN THE
C                           COLLISIONAL-RADIATIVE   SENSE   FROM   THE
C                           PARENT.
C                            => P(TOTAL)/N(IP)          (ERGS SEC-1)
C                             1ST DIMENSION: PARENT METASTABL INDEX
C                             2nd DIMENSION: TEMPERATURE INDEX
C                             3rd DIMENSION: DENSITY     INDEX
C  OUTPUT: (R*8)  PHA(,)  =  EQUILIBRIUM CX POWER COEFFT.
C                            => P(TOTAL)/(DENS*N(1))  (ERGS CM3 SEC-1)
C                             1st DIMENSION: TEMPERATURE INDEX
C                             2nd DIMENSION: DENSITY     INDEX
C
C          (I*4) IM       = METASTABLE LEVEL ARRAY INDEX
C          (I*4) IS       = ORDINARY LEVEL ARRAY INDEX
C          (I*4) IP       = PARENT METASTABLE INDEX
C
C
C
C ROUTINES: NONE
C
C
C AUTHOR:  H. P. SUMMERS, UNIVERSITY OF STRATHCLYDE
C          JA8.08
C          TEL. 0141-553-4196
C
C DATE:    24/05/96
C
C UPDATE:
C
C**********************************************************************
C PUT UNDER S.C.C.S CONTROL:
C
C VERSION: 1.1 DATE: 15/07/96
C MODIFIED: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C     - FIRST PUT UNDER S.C.C.S
C
C
C----------------------------------------------------------------------
C
C----------------------------------------------------------------------
      INTEGER             IMETR(NMET), IN,          IORDR(NORD), IT
      INTEGER             MAXD,        MAXT,        NDDEN,       NDLEV
      INTEGER             NDMET,       NDTEM,       NMET,        NORD
      INTEGER             NPL
      REAL*8              PH(NDTEM,NDDEN,NDMET),    PHA(NDTEM,NDDEN)
      REAL*8              PLA1(NDLEV), RATPIA(NDDEN,NDMET)
      REAL                STVH(NDLEV,NDTEM,NDDEN,NDMET)
      REAL*8              STVHM(NDMET,NDTEM,NDDEN,NDMET)
```

3.60 b8totl: Subroutine b8totl from library adas2xx

```
C
      SUBROUTINE B8TOTL( NDLEV   , NDMET  ,
     &                   NORD    , NMET   ,
     &                   IORDR   , IMETR  , ISTRN ,
     &                   DENSX   ,
     &                   STCKMX  , STACKX ,
     &                   PLA1    , PLBAX  ,
     &                   PLAX    , PLX    ,
     &                   PSAX    , PSX
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: B8TOTL *********************
C
C  PURPOSE: TO CALCULATE TOTAL/SPECIFIC LINE POWERS FOR METASTABLES
C           AND TOTAL/SPECIFIC EQUILIBRIUM LINE POWERS.
C
C           DEVELOPMENT OF B6TOTL.
C
C  NOTE:    A SPECIFIC LINE IS EVALUATED WHICH TERMINATES ON EACH
C           METASTABLE.  EACH IS RESOLVED INTO THE PART DRIVEN BY EACH
C           METASTABLE.  THE EQUILIBRIUM POWER IN EACH OF THESE LINES IS
C           ALSO EVALUATED USING THE EQUILIBRIUM METASTABLE FRACTIONS.
C
C  CALLING PROGRAM:  ADAS208
C
C  SUBROUTINE:
C
C  INPUT : (I*4) NDLEV    = PARAMETER = MAX. NO. OF LEVELS ALLOWED
C  INPUT : (I*4) NDMET    = PARAMETER = MAX. NO. OF METASTABLES ALLOWED
C
C  INPUT : (I*4) NORD     = NUMBER OF ORD. LEVELS (1 <= NORD <= 'NDLEV')
C  INPUT : (I*4) NMET     = NUMBER OF METASTABLES (1 <= NMET <= 'NDMET')
C
C  INPUT : (I*4) IORDR()  = INDEX OF ORDINARY LEVELS IN COMPLETE LEVEL
C                           LIST (ARRAY SIZE = 'NDLEV' )
C  INPUT : (I*4) IMETR()  = INDEX OF METASTABLE IN COMPLETE LEVEL LIST
C                           (ARRAY SIZE = 'NDMET' )
C  INPUT : (I*4)  ISTRN() = SPECIFIC LINE POWER: SELECTED ELECTRON
C                           IMPACT TRANSITION INDEX. (FOR USE WITH
C                           'IE1A()' , 'IE2A()' AND 'AA()' ARRAYS)
C                           WHICH GIVES LARGEST POWER TO METASTABLE
C                            DIMENSION: METASTABLE LINE COUNT INDEX
C  INPUT : (R*8) DENSX    = ELECTRON DENSITY (UNITS: CM-3)
C
C  INPUT : (R*8) STCKMX() = METASTABLE POPULATIONS STACK
C                           AT FIXED TEMPERATURE AND DENSITY.
C                            DIMENSION: METASTABLE INDEX
C  INPUT : (R*4) STACKX(,)= ORDINARY EXCITED LEVEL POPULAT'N DEPENDENCE
C                           ON METASTABLE LEVEL. AT FIXED TEMPERATURE
C                           AND DENSITY.
C                             1st DIMENSION: ORDINARY LEVEL INDEX
C                             2nd DIMENSION: METASTABLE INDEX
C
C  INPUT : (R*8)  PLA1()  = DIRECT LINE POWER LOSS FOR EACH LEVEL.
C                           (UNITS: ERGS SEC-1) (DIMENSION: LEVEL INDEX)
C  INPUT : (R*8)  PLBAX() = HIGH N PROJECTED POWER BASED ON EXCITATIONS
C                           FROM A PARTICULAR METASTABLE TO LEVELS
C                           'IPROJ' UPWARDS. (UNITS: ERGS CM3 SEC-1)
```

```
C                           AT FIXED TEMPERATURE.
C                             DIMENSION: METASTABLE  INDEX
C
C  OUTPUT: (R*8)  PLAX    = TOTAL EQUILIBRIUM LINE POWER COEFFICIENTS.
C                           AT FIXED TEMPERATURE AND DENSITY.
C                            (UNITS: ERGS CM3 SEC-1)
C  OUTPUT: (R*8)  PLX()    = TOTAL LINE POWERS FOR METASTABLES. THIS IS
C                           THE SUM OF ALL EMISSIONS ORGINATING IN THE
C                           COLLISIONAL-RADIATIVE   SENSE   FROM   THE
C                           METASTABLE. AT FIXED TEMPERATURE AND DENSITY
C                           (UNITS: ERGS SEC-1 )
C                             DIMENSION: METASTABLE  INDEX
C
C  OUTPUT: (R*8)  PSAX()   = TOTAL EQUILIBRIUM LINE POWER COEFFICIENTS.
C                           AT FIXED TEMPERATURE AND DENSITY.
C                            (UNITS: ERGS CM3 SEC-1)
C                             DIMENSION: METASTABLE  INDEX
C  OUTPUT: (R*8)  PSX(,)   = TOTAL LINE POWERS FOR METASTABLES. THIS IS
C                           THE SUM OF ALL EMISSIONS ORGINATING IN THE
C                           COLLISIONAL-RADIATIVE   SENSE   FROM   THE
C                           METASTABLE. AT FIXED TEMPERATURE AND DENSITY
C                           (UNITS: ERGS SEC-1 )
C                             1ST. DIMENSION: METASTABLE INDEX FOR LINE
C                             2ND. DIMENSION: METASTABLE INDEX OF DRIVER
C
C          (I*4) IM       = METASTABLE LEVEL ARRAY INDEX
C          (I*4) IS       = ORDINARY LEVEL ARRAY INDEX
C          (I*4) ISL      = SPECIFIC LINE POWER INDEX
C
C
C
C ROUTINES: NONE
C
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/37
C          JET EXT. 5023
C
C DATE:    09/10/90
C
C UPDATE:  20/05/93-P BRIDEN: STACKX ARRAY CHANGED FROM REAL*8 -> REAL*4
C
C UPDATE:  24/05/96  HP SUMMERS - EXTENSION FOR SPECIFIC LINE POWER
C
C**********************************************************************
C PUT UNDER S.C.C.S CONTROL:
C
C VERSION: 1.1 DATE: 16/07/95
C MODIFIED: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C     - FIRST PUT UNDER S.C.C.S
C
C VERSION: 1.2 DATE: 20/07/07
C MODIFIED: Allan Whiteford
C     - Small modification to comments to allow for automatic
C             documentation preparation.
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IMETR(NMET), IORDR(NORD), ISTRN(NDMET)
      INTEGER             NDLEV,       NDMET,       NMET,        NORD
```

```
REAL*8              DENSX,        PLA1(NDLEV), PLAX
REAL*8              PLBAX(NDMET),              PLX(NDMET)
REAL*8              PSAX(NDMET), PSX(NDMET,NDMET)
REAL                STACKX(NDLEV,NDMET)
REAL*8              STCKMX(NDMET)
```

3.61 b8ttyp: Subroutine b8ttyp from library adas2xx

```
C
      SUBROUTINE B8TTYP( NDLEV  , NDMET  , NDTRN  , NPLR  , NPLI  ,
     &                   ITRAN  , TCODE  , I1A    , I2A   , AVAL  ,
     &                   ICNTE  , ICNTP  , ICNTR  , ICNTH , ICNTI ,
     &                   ICNTL  , ICNTS  ,
     &                   IETRN  , IPTRN  , IRTRN  , IHTRN , IITRN ,
     &                   ILTRN  , ISTRN  ,
     &                                     IE1A   , IE2A  , AA    ,
     &                                     IP1A   , IP2A  ,
     &                                     IA1A   , IA2A  , AUGA  ,
     &                                     IL1A   , IL2A  , WVLA  ,
     &                                     IS1A   , IS2A  , LSS04A
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: B8TTYP *********************
C
C  PURPOSE:  TO SORT TRANSITION ARRAYS INTO SEVEN TRANSITION/RECOMB TYPES
C
C  CALLING PROGRAM: ADAS208
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  NDLEV   = MAXIMUM NUMBER OF LEVELS THAT CAN BE READ
C  INPUT : (I*4)  NDMET   = MAXIMUM NUMBER OF METASTABLES
C  INPUT : (I*4)  NDTRN   = MAXIMUM NUMBER OF TRANS. THAT CAN BE READ
C  INPUT : (I*4)  NPLR    = NO. OF ACTIVE METASTABLES OF (Z+1) ION
C  INPUT : (I*4)  NPLI    = NO. OF ACTIVE METASTABLES OF (Z-1) ION
C
C  INPUT : (I*4)  ITRAN   = INPUT DATA FILE: NUMBER OF TRANSITIONS
C  INPUT : (C*1)  TCODE() = TRANSITION: DATA TYPE POINTER:
C                           ' ' => Electron Impact   Transition
C                           'P' => Proton   Impact   Transition
C                           'H' => Charge   Exchange Recombination
C                           'R' => Free     Electron Recombination
C                           'I' => Electron Impact Ionisation to z
C                           'L' => Satellites from DR Recombination
C                           'S' => Electron Impact Ionisation to z+1
C  INPUT : (I*4)  I1A()   = TRANSITION:
C                            LOWER ENERGY LEVEL INDEX (CASE ' ' & 'P')
C                            PARENT ENERGY LEVEL INDEX(CASE 'H' & 'R')
C                                                     (         & 'L')
C                            FINAL PARENT LEVEL INDEX (CASE 'S')
C  INPUT : (I*4)  I2A()   = TRANSITION:
C                            UPPER ENERGY LEVEL INDEX (CASE ' ' & 'P')
C                            CAPTURING    LEVEL INDEX (CASE 'H' & 'R')
C                                                     (         & 'L')
C                            IONISING     LEVEL INDEX (CASE 'S')
C  INPUT : (R*8)  AVAL()  = TRANSITION:
C                            A-VALUE (SEC-1)          (CASE ' ')
C                            NEUTRAL BEAM ENERGY      (CASE 'H')
C                            AUGER VALUE(SEC-1)       (CASE 'R')
C                            PARENT WAVLENGTH (A)     (CASE 'L')
C                            NOT USED                 (CASE 'P' & 'S')
C
C  OUTPUT: (I*4)  ICNTE   = NUMBER OF ELECTRON IMPACT TRANSITIONS INPUT
C  OUTPUT: (I*4)  ICNTP   = NUMBER OF PROTON IMPACT TRANSITIONS INPUT
C  OUTPUT: (I*4)  ICNTR   = NUMBER OF FREE ELECTRON RECOMBINATIONS INPUT
C  OUTPUT: (I*4)  ICNTH   = NO. OF CHARGE EXCHANGE RECOMBINATIONS INPUT
```

```
C  OUTPUT: (I*4)  ICNTI   = NO. OF IONISATIONS TO Z INPUT
C  OUTPUT: (I*4)  ICNTL   = NO. OF SATELLITE DR RECOMBINATIONS INPUT
C  OUTPUT: (I*4)  ICNTS   = NO. OF IONISATIONS TO Z+1 INPUT
C
C  OUTPUT: (I*4)  IETRN() = ELECTRON IMPACT TRANSITION:
C                           INDEX VALUES IN MAIN TRANSITION ARRAYS WHICH
C                           REPRESENT ELECTRON IMPACT TRANSITIONS.
C  OUTPUT: (I*4)  IPTRN() = PROTON IMPACT TRANSITION:
C                           INDEX VALUES IN MAIN TRANSITION ARRAYS WHICH
C                           REPRESENT PROTON IMPACT TRANSITIONS.
C  OUTPUT: (I*4)  IRTRN() = FREE ELECTRON RECOMBINATION:
C                           INDEX VALUES IN MAIN TRANSITION ARRAYS WHICH
C                           REPRESENT FREE ELECTRON RECOMBINATIONS.
C  OUTPUT: (I*4)  IHTRN() = CHARGE EXCHANGE RECOMBINATION:
C                           INDEX VALUES IN MAIN TRANSITION ARRAYS WHICH
C                           REPRESENT CHARGE EXCHANGE RECOMBINATIONS.
C  OUTPUT: (I*4)  IITRN() = ELECTRON IMPACT IONISATION:
C                           INDEX VALUES IN MAIN TRANSITION ARRAYS WHICH
C                           REPRESENT IONISATIONS FROM LOWER STAGE ION.
C  OUTPUT: (I*4)  ILTRN() = SATELLITE DR RECOMBINATION:
C                           INDEX VALUES IN MAIN TRANSITION ARRAYS WHICH
C                           REPRESENT SATELLITE DR RECOMBINATIONS.
C  OUTPUT: (I*4)  ISTRN() = ELECTRON IMPACT IONISATION:
C                           INDEX VALUES IN MAIN TRANSITION ARRAYS WHICH
C                           REPRESENT IONISATIONS TO UPPER STAGE ION.
C
C  OUTPUT: (I*4)  IE1A()  = ELECTRON IMPACT TRANSITION:
C                            LOWER ENERGY LEVEL INDEX
C  OUTPUT: (I*4)  IE2A()  = ELECTRON IMPACT TRANSITION:
C                            UPPER ENERGY LEVEL INDEX
C  OUTPUT: (R*8)  AA()    = ELECTRON IMPACT TRANSITION: A-VALUE (SEC-1)
C
C
C  OUTPUT: (I*4)  IP1A()  = PROTON IMPACT TRANSITION:
C                            LOWER ENERGY LEVEL INDEX
C  OUTPUT: (I*4)  IP2A()  = PROTON IMPACT TRANSITION:
C                            UPPER ENERGY LEVEL INDEX
C
C  OUTPUT: (I*4)  IA1A()  = AUGER TRANSITION:
C                            PARENT ENERGY LEVEL INDEX
C  OUTPUT: (I*4)  IA2A()  = AUGER TRANSITION:
C                            RECOMBINED ION ENERGY LEVEL INDEX
C  OUTPUT: (R*8)  AUGA()  = AUGER TRANSITION: AUG-VALUE (SEC-1)
C                            RECOMBINED ION ENERGY LEVEL INDEX
C  OUTPUT: (I*4)  IL1A()  = SATELLITE DR TRANSITION:
C                            RECOMBINING ION  INDEX
C  OUTPUT: (I*4)  IL2A()  = SATELLITE DR TRANSITION:
C                            RECOMBINED ION INDEX
C  OUTPUT: (R*8)  WVLA()  = SATELLITE DR TRANSITION: PARENT WVLGTH.(A)
C                            DR SATELLITE LINE INDEX
C  OUTPUT: (I*4)  IS1A()  = IONISING TRANSITION:
C                            IONISED ION  INDEX
C  OUTPUT: (I*4)  IS2A()  = IONISING TRANSITION:
C                            IONISING ION INDEX
C  OUTPUT: (L*4)  LSS04A(,)= .TRUE. => IONIS. RATE SET IN ADF04 FILE:
C                            .FALSE.=> NOT SET IN ADF04 FILE
C                            1ST DIM: LEVEL INDEX
C                            2ND DIM: PARENT METASTABLE INDEX
C
C          (I*4)  I       = GENERAL USE.
C
```

```
C
C ROUTINES: NONE
C
C AUTHOR:  HP SUMMERS (REVISION OF BXTTYP BY PE BRIDEN)
C          K1/1/57
C          JET EXT. 4941
C
C DATE  :  11/06/92
C
C**********************************************************************
C UNIX-IDL PORT:
C
C AUTHOR: DAVID H BROOKS, UNIVERSITY OF STRATHCLYDE
C
C DATE: UNKNOWN
C
C**********************************************************************
C PUT UNDER SCCS CONTROL:
C
C VERSION: 1.1 DATE: 10/05/96
C MODIFIED: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C     - FIRST PUT UNDER SCCS
C
C VERSION: 1.2 DATE: 13/09/99
C MODIFIED: HUGH SUMMERS, UNIVERSITY OF STRATHCLYDE
C     - ADDED DETECTION OF L-LINES AND S-LINES
C
C----------------------------------------------------------------------
C----------------------------------------------------------------------
      CHARACTER           TCODE(NDTRN)
      INTEGER             I1A(NDTRN),  I2A(NDTRN),  IA1A(NDTRN)
      INTEGER             IA2A(NDTRN), ICNTE,       ICNTH,       ICNTI
      INTEGER             ICNTL,       ICNTP,       ICNTR,       ICNTS
      INTEGER             IE1A(NDTRN), IE2A(NDTRN), IETRN(NDTRN)
      INTEGER             IHTRN(NDTRN),             IITRN(NDTRN)
      INTEGER             IL1A(NDLEV), IL2A(NDLEV), ILTRN(NDTRN)
      INTEGER             IP1A(NDTRN), IP2A(NDTRN), IPTRN(NDTRN)
      INTEGER             IRTRN(NDTRN),             IS1A(NDLEV)
      INTEGER             IS2A(NDLEV), ISTRN(NDTRN),             ITRAN
      INTEGER             NDLEV,       NDMET,       NDTRN,       NPLI
      INTEGER             NPLR
      LOGICAL             LSS04A(NDLEV,NDMET)
      REAL*8              AA(NDTRN),   AUGA(NDTRN), AVAL(NDTRN)
      REAL*8              WVLA(NDLEV)
```

3.62 b8winf: Subroutine b8winf from library adas2xx

```
C
      SUBROUTINE B8WINF( IUNIT  , LADF10 , DATE   , USER    ,
     &                   NDLEV  ,
     &                   DSNINC , DSFULL , DSNEXP ,
     &                   IZ0    , IZ1    ,
     &                   IL     , NMET   , NPL    , IBSELA  ,
     &                   LRSEL  , LISEL  , LHSEL  , LIOSEL  ,
     &                   LPSEL  , LZSEL  , LNSEL  , LNORM   ,
     &                   LSSETA , LSS04A
     &                 )

C-----------------------------------------------------------------------
C
C  ******************** FORTRAN77 SUBROUTINE: B8WINF ******************
C
C  PURPOSE:  Write information section of adf10, adf13 and adf15 files.
C
C
C  CALLING PROGRAM: ADAS208 - b8wrmc, b8wr11, b8wr12
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  IUNIT    = OUTPUT UNIT NUMBER
C  INPUT : (I*4)  LADF10   = .TRUE. IF WRITING COMMENTS TO ADF10 FILE.
C  INPUT : (C*8)  DATE     = CURRENT DATE.
C  INPUT : (C*30) USER     = USER IDENTIFIER
C  INPUT : (I*4)  NDLEV    = MAXIMUM NUMBER OF LEVELS ALLOWED
C  INPUT : (I*4)  NDMET    = MAXIMUM NUMBER OF METASTABLES ALLOWED
C  INPUT : (C*80) DSNINC   = INPUT COPASE DATA SET NAME (IN QUOTES).
C  INPUT : (C*80) DSFULL   = INPUT SZD  DATA SET NAME (IN QUOTES).
C  INPUT : (C*80) DSNEXP   = INPUT EXPANSION FILE
C  INPUT : (I*4)  IZ0      =         NUCLEAR CHARGE READ
C  INPUT : (I*4)  IZ1      = RECOMBINING ION CHARGE READ
C                            (NOTE: IZ1 SHOULD EQUAL IZ+1)
C  INPUT : (I*4)  IL       = NUMBER OF ENERGY LEVELS
C  INPUT : (I*4)  NMET     = NUMBER OF METASTABLES LEVELS: 1<=NMET<=NDMET
C  INPUT : (I*4)  NPL      = NO. OF METASTABLES OF (Z+1) ION ACCESSED
C                              BY EXCITED STATE IONISATION IN COPASE
C                              FILE WITH IONISATION POTENTIALS GIVEN
C                              ON THE FIRST DATA LINE
C  INPUT : (I*4)  IBSELA(,)=IONISATION DATA BLOCK SELECTOR INDICES
C                           1ST DIMENSION - (Z)   ION METASTABLE COUNT
C                           2ND DIMENSION - (Z+1) ION METASTABLE COUNT
C  INPUT : (L*4)  LRSEL    = .TRUE.  -  RECOMB  OF (Z+1) ION ACTIVE
C                            .FALSE. -  RECOMB. OF (Z+1) ION INACTIVE
C  INPUT : (L*4)  LISEL    = .TRUE.  -  IONIS.  OF (Z-1) ION ACTIVE
C                            .FALSE. -  IONIS.  OF (Z-1) ION INACTIVE
C  INPUT : (L*4)  LHSEL    = .TRUE.  -  CX REC. OF (Z+1) ION ACTIVE
C                            .FALSE. -  CX REC. OF (Z+1) ION INACTIVE
C  INPUT : (L*4)  LIOSEL   = .TRUE.  -  IONIS.  OF (Z) ION ACTIVE
C                            .FALSE. -  IONIS.  OF (Z) ION INACTIVE
C  INPUT : (L*4)  LPSEL    = .TRUE.  => INCLUDE PROTON COLLISIONS
C                          = .FALSE. =>DO NOT INCLUDE PROTON COLLISIONS
C  INPUT : (L*4)  LZSEL    = .TRUE.  => SCALE PROTON COLLISIONS WITH
C                                       PLASMA Z EFFECTIVE'ZEFF'.
C                          = .FALSE. => DO NOT SCALE PROTON COLLISIONS
C                                       WITH PLASMA Z EFFECTIVE 'ZEFF'.
C                            (ONLY USED IF 'LPSEL=.TRUE.')
C  INPUT : (L*4)  LNSEL    = .TRUE.  => INCLUDE PROJECTED BUNDLE-N DATA
```

```
C                                        FROM DATAFILE IF AVAILABLE
C                          = .FALSE. => DO NOT INCLUDE PROJECTED
C                                       BUNDLE-N DATA
C  INPUT : (L*4)  LNORM    = .TRUE.  => IF NMET=1 THEN VARIOUS
C                                       IONISATION OUTPUT FILE
C                                       NORMALISED TO STAGE TOT.POPULATN.
C                                       (** NORM TYPE = T)
C                          = .FALSE. => OTHERWISE NORMALISE TO IDENTIFIED
C                                       METASTABLE POPULATIONS.
C                                        (** NORM TYPE = M)
C  INPUT : (L*4)  LSSETA(,)= .TRUE.  => MET. IONIS RATE SET IN B8GETS
C                           .FALSE. => MET. IONIS RATE NOT SET IN B8GETS
C                            1ST DIMENSION: (Z) ION METASTABLE INDEX
C                            2ND DIMENSION: (Z+1) ION METASTABLE INDEX
C  INPUT : (L*4)  LSS04A(,)= .TRUE.  => IONIS. RATE SET IN ADF04 FILE:
C                           .FALSE. => NOT SET IN ADF04 FILE
C                            1ST DIM: LEVEL INDEX
C                            2ND DIM: PARENT METASTABLE INDEX
C
C
C  INTERNAL:
C
C          (L*4)  LION     = .TRUE.  => SOME/ALL MET. IONIS RATE IS SET
C          (L*4)  LIONALL  = .TRUE.  => ALL MET. IONIS RATES ARE SET
C          (L*4)  LIONA    = .TRUE.  => ALL MET. IONIS RATES FROM ADF04 FILE
C          (L*4)  LIONOV   = .TRUE.  => SOME/ALL MET. IONIS RATES TAKEN
C                                       FROM ADF07 FILE
C
C
C  ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          XXSLEN     ADAS     FINDS LENGTH OF NON-BLANK STRINGS
C
C
C AUTHOR:  Martin O'Mullane
C
C DATE:    8/10/99
C
C VERSION:  1.1                                  DATE: 8/10/99
C MODIFIED: Martin O'Mullane
C           - First version
C VERSION:  1.2                                  DATE: 21/03/00
C MODIFIED: Martin O'Mullane
C - Removed NDMET from input parameter list
C
C-----------------------------------------------------------------------
      CHARACTER*8         DATE
      CHARACTER*80        DSFULL,      DSNEXP,      DSNINC
      CHARACTER*30        USER
      INTEGER             IBSELA(NDMET,NDMET),      IL,          IUNIT
      INTEGER             IZ0,         IZ1,         NDLEV,       NMET
      INTEGER             NPL
      LOGICAL             LADF10,      LHSEL,       LIOSEL,      LISEL
      LOGICAL             LNORM,       LNSEL,       LPSEL,       LRSEL
      LOGICAL             LSS04A(NDLEV,NDMET),      LSSETA(NDMET,NDMET)
      LOGICAL             LZSEL
```

3.63 b8wrmc: Subroutine b8wrmc from library adas2xx

```
C
      SUBROUTINE B8WRMC( IUNIT  , IUNT14 , IUNT15 , IUNT16 , IUNT17 ,
     &                   IUNT18 , IUNT19 , IUNT20 , IUNT21 ,
     &                   IUNT22 , IUNT23 ,
     &                   DSNINC , DSFULL , DSNEXP ,
     &                   TITLED , DATE   , USER   ,
     &                   NDLEV  , NDTEM  , NDDEN  , NDMET  ,
     &                   LNORM  , IZ     , IZ0    , IZ1    ,
     &                   IBSELA , BWNOA  , PRTWTA ,
     &                   IL     , NMET   , NORD   , IMETR  ,
     &                   IA     , ISA    , ILA    , XJA    ,
     &                   CSTRGA , WA     ,
     &                   MAXT   , MAXD   , TEVA   , DENSA  ,
     &                   NPL    , NPLR   , NPL3   , NPLI   , CPRTA  ,
     &                   LRSEL  , LISEL  , LHSEL  , LIOSEL ,
     &                   LPSEL  , LZSEL  , LNSEL  , FVCRED ,
     &                   FVRRED , FVIRED , FVHRED , FVIONR ,
     &                   FVCRPR , PL     , PH     , PS     , SWVLN  ,
     &                   PR     ,
     &                   RATPIA , RATMIA , STACK  , STCKM  ,
     &                   LSSETA , LSS04A
     &                 )
C-----------------------------------------------------------------------
C
C  ******************** FORTRAN77 SUBROUTINE: B8WRMC ***************** *
C
C  PURPOSE:  TO OUTPUT DATA TO GENERALISED COLLISIONAL RADIATIVE
C            COEFFICIENT PASSING FILE MASTER.PASS
C            FINAL STORAGE IS EXPECTED TO BE IN MASTER CONDENSED FILES
C
C
C  CALLING PROGRAM: ADAS208
C
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  IUNIT   = OUTPUT UNIT NUMBER FOR GCR INFORMATION
C  INPUT : (I*4)  IUNT14  = OUTPUT UNIT NUMBER FOR ACD DATA
C  INPUT : (I*4)  IUNT15  = OUTPUT UNIT NUMBER FOR SCD DATA
C  INPUT : (I*4)  IUNT16  = OUTPUT UNIT NUMBER FOR CCD DATA
C  INPUT : (I*4)  IUNT17  = OUTPUT UNIT NUMBER FOR QCD DATA
C  INPUT : (I*4)  IUNT18  = OUTPUT UNIT NUMBER FOR XCD DATA
C  INPUT : (I*4)  IUNT19  = OUTPUT UNIT NUMBER FOR PRB DATA
C  INPUT : (I*4)  IUNT20  = OUTPUT UNIT NUMBER FOR PRC DATA
C  INPUT : (I*4)  IUNT21  = OUTPUT UNIT NUMBER FOR PLT DATA
C  INPUT : (I*4)  IUNT22  = OUTPUT UNIT NUMBER FOR PLS DATA
C  INPUT : (I*4)  IUNT23  = OUTPUT UNIT NUMBER FOR MET DATA
C  INPUT : (C*80) DSNINC  = INPUT COPASE DATA SET NAME (IN QUOTES).
C  INPUT : (C*80) DSFULL  = INPUT SZD  DATA SET NAME (IN QUOTES).
C  INPUT : (C*80) DSNEXP  = INPUT EXPANSION FILE
C  INPUT : (C*3)  TITLED  = ELEMENT SYMBOL.
C  INPUT : (C*8)  DATE    = CURRENT DATE.
C  INPUT : (C*30) USER    = USER IDENTIFIER
C
C  INPUT : (I*4)  NDLEV   = MAXIMUM NUMBER OF LEVELS ALLOWED
C  INPUT : (I*4)  NDTEM   = MAXIMUM NUMBER OF TEMPERATURES ALLOWED
C  INPUT : (I*4)  NDDEN   = MAXIMUM NUMBER OF DENSITIES ALLOWED
C  INPUT : (I*4)  NDMET   = MAXIMUM NUMBER OF METASTABLES ALLOWED
C
```

```
C  INPUT : (I*4)  IZ      =  RECOMBINED ION CHARGE READ
C  INPUT : (I*4)  IZ0     =         NUCLEAR CHARGE READ
C  INPUT : (I*4)  IZ1     = RECOMBINING ION CHARGE READ
C                           (NOTE: IZ1 SHOULD EQUAL IZ+1)
C  INPUT : (I*4)  IBSELA(,)=IONISATION DATA BLOCK SELECTOR INDICES
C                           1ST DIMENSION - (Z)   ION METASTABLE COUNT
C                           2ND DIMENSION - (Z+1) ION METASTABLE COUNT
C  INPUT : (R*8)  BWNOA() = IONISATION POTENTIALS TO (Z+1) METAS.(CM-1)
C  INPUT : (R*8)  PRTWTA()= STATISTICAL WEIGHTS OF (Z+1) METASTABLES
C
C  INPUT : (I*4)  IL      = NUMBER OF ENERGY LEVELS
C  INPUT : (L*4)  LNORM   =.TRUE.  => IF NMET=1 THEN VARIOUS
C                                     IONISATION OUTPUT FILE
C                                     NORMALISED TO STAGE TOT.POPULATN.
C                                     (** NORM TYPE = T)
C                         =.FALSE. => OTHERWISE NORMALISE TO IDENTIFIED
C                                     METASTABLE POPULATIONS.
C                                      (** NORM TYPE = M)
C
C  INPUT : (I*4)  NMET    = NUMBER OF METASTABLES LEVELS: 1<=NMET<=NDMET
C  INPUT : (I*4)  NORD    = NUMBER OF ORDINARY LEVELS ('IL' - 'NMET')
C  INPUT : (I*4)  IMETR() = INDEX OF (Z) METAS. IN COMPLETE LEVEL LIST
C
C  INPUT : (I*4)  IA()    = ENERGY LEVEL INDEX NUMBER
C  INPUT : (I*4)  ISA()   = MULTIPLICITY FOR LEVEL 'IA()'
C                           NOTE: (ISA-1)/2 = QUANTUM NUMBER (S)
C  INPUT : (I*4)  ILA()   = QUANTUM NUMBER (L) FOR LEVEL 'IA()'
C  INPUT : (R*8)  XJA()   = QUANTUM NUMBER (J-VALUE) FOR LEVEL 'IA()'
C                           NOTE: (2*XJA)+1 = STATISTICAL WEIGHT
C  INPUT : (R*8)  WA()    = ENERGY RELATIVE TO LEVEL 1 (CM-1)
C                           DIMENSION: LEVEL INDEX
C  INPUT : (R*8)  AA()    = ELECTRON IMPACT TRANSITION: A-VALUE (SEC-1)
C
C  INPUT : (R*8)  SGRDA(,,)=GROUND & METASTABLE IONISATION RATE
C                           COEFFICIENTS  FROM SZD FILES (CM3 SEC-1)
C                           1ST DIMENSION: TEMPERATURE INDEX
C                           2ND DIMENSION: (Z)   ION METASTABLE INDEX
C                           3RD DIMENSION: (Z+1) ION METASTABLE INDEX
C  INPUT : (C*18) CSTRGA()= NOMENCLATURE/CONFIGURATION FOR LEVEL 'IA()'
C  INPUT : (I*4)  MAXT    = NUMBER OF INPUT TEMPERATURES ( 1 -> 'NDTEM')
C  INPUT : (I*4)  MAXD    = NUMBER OF INPUT DENSITIES ( 1 -> 'NDDEN')
C  INPUT : (R*8)  TEA()   = ELECTRON TEMPERATURES (UNITS: KELVIN)
C  INPUT : (R*8)  DENSA() = ELECTRON DENSITIES  (UNITS: CM-3)
C
C  INPUT : (I*4)  NPL      = NO. OF METASTABLES OF (Z+1) ION ACCESSED
C                            BY EXCITED STATE IONISATION IN COPASE
C                            FILE WITH IONISATION POTENTIALS GIVEN
C                            ON THE FIRST DATA LINE
C  INPUT : (I*4)  NPLR     = NO. OF ACTIVE METAS. FOR RECOM OF (Z+1) ION
C  INPUT : (I*4)  NPL3     = NO. OF ACTIVE METAS. FOR RE+3B OF (Z+1) ION
C  INPUT : (I*4)  NPLI     = NO. OF ACTIVE METASTABLES OF (Z-1) ION
C
C  INPUT : (L*4)  LRSEL    = .TRUE.  -  RECOMB  OF (Z+1) ION ACTIVE
C                            .FALSE. -  RECOMB. OF (Z+1) ION INACTIVE
C  INPUT : (L*4)  LISEL    = .TRUE.  -  IONIS.  OF (Z-1) ION ACTIVE
C                            .FALSE. -  IONIS.  OF (Z-1) ION INACTIVE
C  INPUT : (L*4)  LHSEL    = .TRUE.  -  CX REC. OF (Z+1) ION ACTIVE
C                            .FALSE. -  CX REC. OF (Z+1) ION INACTIVE
C  INPUT : (L*4)  LIOSEL   = .TRUE.  -  IONIS.  OF (Z) ION ACTIVE
C                            .FALSE. -  IONIS.  OF (Z) ION INACTIVE
C  INPUT : (L*4)  LPSEL    = .TRUE.  => INCLUDE PROTON COLLISIONS
```

```
C                            = .FALSE. =>DO NOT INCLUDE PROTON COLLISIONS
C  INPUT : (L*4)  LZSEL      = .TRUE.  => SCALE PROTON COLLISIONS WITH
C                                         PLASMA Z EFFECTIVE'ZEFF'.
C                            = .FALSE. => DO NOT SCALE PROTON COLLISIONS
C                                         WITH PLASMA Z EFFECTIVE 'ZEFF'.
C                           (ONLY USED IF 'LPSEL=.TRUE.')
C  INPUT : (L*4)  LNSEL      = .TRUE.  => INCLUDE PROJECTED BUNDLE-N DATA
C                                          FROM DATAFILE IF AVAILABLE
C                            = .FALSE. => DO NOT INCLUDE PROJECTED
C                                         BUNDLE-N DATA
C
C  INPUT : (R*8) FVCRED(,,,) = (Z)-(Z) CROSS GEN. COLL. RAD. COEFFTS.
C                              1ST DIMENSION: (Z) METASTABLE INDEX
C                              2ND DIMENSION: (Z) METASTABLE INDEX
C                              3RD DIMENSION: TEMPERATURE INDEX
C                              4TH DIMENSION: DENSITY INDEX
C  INPUT : (R*8) FVRRED(,,,) = (Z+1)-(Z) RECOM GEN. COLL. RAD. COEFFTS.
C                              1ST DIMENSION: (Z) METASTABLE INDEX
C                              2ND DIMENSION: (Z+1) METASTABLE INDEX
C                              3RD DIMENSION: TEMPERATURE INDEX
C                              4TH DIMENSION: DENSITY INDEX
C  INPUT : (R*8) FVIRED(,,,) = (Z-1)-(Z) IONIS GEN. COLL. RAD. COEFFTS.
C                              1ST DIMENSION: (Z) METASTABLE INDEX
C                              2ND DIMENSION: (Z-1) METASTABLE INDEX
C                              3RD DIMENSION: TEMPERATURE INDEX
C                              4TH DIMENSION: DENSITY INDEX
C  INPUT : (R*8) FVHRED(,,,) = (Z+1)-(Z) CX R. GEN. COLL. RAD. COEFFTS.
C                              1ST DIMENSION: (Z) METASTABLE INDEX
C                              2ND DIMENSION: (Z+1) METASTABLE INDEX
C                              3RD DIMENSION: TEMPERATURE INDEX
C                              4TH DIMENSION: DENSITY INDEX
C  INPUT : (R*8) FVIONR(,,,) = (Z)-(Z+1) IONIS GEN. COLL. RAD. COEFFTS.
C                              1ST DIMENSION: (Z) METASTABLE INDEX
C                              2ND DIMENSION: (Z+1) METASTABLE INDEX
C                              3RD DIMENSION: TEMPERATURE INDEX
C                              4TH DIMENSION: DENSITY INDEX
C  INPUT : (R*8) FVCRPR(,,,) = (Z+1)-(Z+1) CROSS COLL. RAD. COEFFTS.
C                              1ST DIMENSION: (Z+1) METASTABLE INDEX
C                                    FINAL STATE
C                              2ND DIMENSION: (Z+1) METASTABLE INDEX
C                                    INITIAL STATE
C                              3RD DIMENSION: TEMPERATURE INDEX
C                              4TH DIMENSION: DENSITY INDEX
C  INPUT : (R*8) PL(,,)       = TOTAL LINE POWER COEFFICIENTS
C                              1ST DIMENSION: (Z) METASTABLE INDEX
C                              2ND DIMENSION: TEMPERATURE INDEX
C                              3RD DIMENSION: DENSITY INDEX
C                                  UNITS: ERG SEC-1
C  INPUT : (R*8) PH(,,)       = CX RECOMBINATION POWER COEFFICIENTS
C                              1ST DIMENSION: TEMPERATURE INDEX
C                              2ND DIMENSION: DENSITY INDEX
C                              3RD DIMENSION: (Z+1) PARENT METAS. INDEX
C                                  UNITS: ERG SEC-1
C  INPUT : (R*8) PS(,,,)      = SPECIFIC LINE POWER COEFFICIENTS
C                              1ST DIMENSION: METASTABLE LINE INDEX
C                              2ND DIMENSION: (Z) METASTABLE INDEX
C                              3RD DIMENSION: TEMPERATURE INDEX
C                              4TH DIMENSION: DENSITY INDEX
C                                  UNITS: ERG SEC-1
C  INPUT : (R*8) SWVLN()      = WAVELENGTHS (ANGSTROM) OF SPECIFIC LINES
C                              1ST DIMENSION: METASTABLE LINE INDEX
```

```
C  INPUT : (R*8)  PR(,,)    = RECOM/BREMS. COEFFT (ERG S-1)
C                             1ST DIM: PARENT INDEX
C                             2ND DIM: TEMPERATURE INDEX
C                             3RD DIM: DENSITY INDEX
C  INPUT : (R*8)  RATPIA(,) = RATIO ( N(Z+1)/N(Z)  STAGE ABUNDANCIES )
C                             1ST DIMENSION: TEMP/DENS INDEX
C                             2ND DIMENSION: PARENT INDEX
C  INPUT : (R*8)  RATMIA(,) = RATIO ( N(Z-1)/N(Z)  STAGE ABUNDANCIES )
C                             1ST DIMENSION: TEMP/DENS INDEX
C                             2ND DIMENSION: PARENT INDEX
C
C  INPUT : (R*4)  STACK(,,,)= POPULATION DEPENDENCE
C                             1ST DIMENSION: ORDINARY LEVEL INDEX
C                             2ND DIMENSION: METASTABLE INDEX
C                             3RD DIMENSION: TEMPERATURE INDEX
C                             4TH DIMENSION: DENSITY INDEX
C  INPUT : (R*8)  STCKM(,,) = METASTABLE POPULATIONS STACK
C                             1st DIMENSION: METASTABLE INDEX
C                             2nd DIMENSION: TEMPERATURE INDEX
C                             3rd DIMENSION: DENSITY INDEX
C  INPUT : (L*4)  LSSETA(,) = .TRUE.  - MET. IONIS RATE SET IN B8GETS
C                             .FALSE.- MET. IONIS RATE NOT SET IN B8GETS
C                              1ST DIMENSION: (Z) ION METASTABLE INDEX
C                              2ND DIMENSION: (Z+1) ION METASTABLE INDEX
C  INPUT : (L*4)  LSS04A(,) = .TRUE. => IONIS. RATE SET IN ADF04 FILE:
C                             .FALSE.=> NOT SET IN ADF04 FILE
C                              1ST DIM: LEVEL INDEX
C                              2ND DIM: PARENT METASTABLE INDEX
C
C          (R*8) DE        = ENERGY FOR TRANSITION ( CM-1)
C                            (IONIS. POT. FOR IONISATION COEFFTS.
C                             EXCIT. ENR. FOR EXCITATION COEFFTS.)
C
C          (I*4) I         = GENERAL USE
C          (I*4) IP        = GENERAL USE
C          (I*4) J         = GENERAL USE
C          (I*4) K         = GENERAL USE
C          (I*4) L         = GENERAL USE
C          (R*8) Z1        = RECOMBINING ION CHARGE
C          (R*8) DUM1      = GENERAL USE
C          (R*8) DUM2      = GENERAL USE
C          (R*8) DUM3      = GENERAL USE
C          (R*8) TR()      = REDUCED TEMPERATUTES ( TE(K) / Z1*Z1)
C          (R*8) DR()      = REDUCED DENSITIES    ( NE/ Z1**7)
C          (R*8) SUM()     = GENERAL USE IN RENORMALISATION
C          (R*8) FMULT()   = GENERAL USE IN RENORMALISATION
C
C
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          B8CORR     ADAS      'FIXES' LOW TE PROBLEM IN REC. DATA
C          B8WINF     ADAS      DETERMINES IONIS. SOURCE AND WRITES
C                               COMMENT BLOCK
C
C
C AUTHOR:  H. P. SUMMERS
C          K1/1/57
C          JET EXT. 4941
C
```

```
C DATE:    24/06/92
C
C UPDATE:  13/08/93  HP SUMMERS  - INCLUDE NORMALISING TO TOTALS WHEN
C                                  (LNORM. AND.(NMET.EQ.1)
C-----------------------------------------------------------------------
C UNIX-IDL PORT:
C
C AUTHOR: DAVID H BROOKS, UNIVERSITY OF STRATHCLYDE
C
C DATE: UNKNOWN
C
C UPDATE:  04/03/96  HP SUMMERS  - OUTPUT C-R DATA TO SEPARATE FILES
C                                  INCLUDE METASTABLE FRACTION FILE
C                                  USE IUNIT FOR INFORMATION.
C UPDATE:  03/05/96  DH BROOKS   - CHANGED DSNINC & DSNEXP TO 80
C                                  CHARACTERS. ALTERED FORMATS 1003
C                                  & 2042 TO ACCOMODATE.
C UPDATE:  13/05/96  HP SUMMERS  - CORRECT TITLE LINE ON QCD208.PASS
C                                  FILE TO GIVE CORRECT JGRD, IGRD
C                                  NAMES.
C UPDATE:  24/05/96  HP SUMMERS  - ADDED SPECIFIC LINE DATA, PS AND
C                                  SWVLN TO PARAMETER LIST
C UPDATE:  03/06/96  HP SUMMERS  - ADDED CX RECOMBINATION DATA, PH
C
C UPDATE:  23/07/96  HP SUMMERS  - TIDY UP NAMES IN OUTPUT FILES FOR
C                                  CONSISTENCY
C
C
C UPDATE:  09/03/98  HP SUMMERS  - ADDED PRB TO DATA PASSED FROM THE
C                                  PROJECTION MATRICES AND GIVEN AS
C                                  OUTPUT FROM ADAS208
C-----------------------------------------------------------------------
C PUT UNDER SCCS CONTROL:
C
C VERSION: 1.1 DATE: 10/05/96
C MODIFIED: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C     - FIRST PUT UNDER SCCS
C
C VERSION: 1.2 DATE: 13/05/96
C MODIFIED: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C     - INCREASED SIZE OF DSFULL TO 80
C
C VERSION: 1.3 DATE: 14-05-96
C MODIFIED: WILLIAM OSBORN
C     REARRANGED ARGUMENTS TO STAY UNDER
C     LIMIT OF 20 CONTINUATION CHARACTERS AT ARCETRI AND GARCHING
C
C VERSION: 1.4 DATE: 15-07-96
C MODIFIED: WILLIAM OSBORN
C     ADDED HUGH'S CORRECTIONS DATED 13/05/96, 24/05/96 AND
C     03/06/96 ABOVE
C
C VERSION: 1.5 DATE: 30-09-96
C MODIFIED: WILLIAM OSBORN
C     ADDED HUGH'S CORRECTIONS DATED 23/07/96 ABOVE
C
C VERSION: 1.6 DATE: 18/10/96
C MODIFIED: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C           - ADDED CHECK FOR INDEX2.EQ.0 IN STRING PROCESSING
C
C VERSION: 1.7 DATE: 01/12/97
```

```
C MODIFIED: RICHARD MARTIN
C - FIXED BUG IN WRITING OUT OF TEMPERATURES IN PLT208.PASS,
C   PLS208.PASS & MET208.PASS
C
C VERSION: 1.8 DATE: 09/03/98
C MODIFIED: HUGH SUMMERS
C    - ADDED PRB TO DATA PASSED FROM THE  PROJECTION MATRICES
C   AND GIVEN AS OUTPUT FROM ADAS208.
C
C VERSION:  1.9                                  DATE: 2/09/99
C MODIFIED: Martin O'Mullane
C           - Pass in real name of author rather than uid.
C           - Removed nulls on output files
C           - Changed delimeter from () to _. in getting FILEMEM
C           - Incorrect logic in writing xcd files. Moved ENDIF to
C             end of inner DO loop. Don't write info blocks for
C             data which does not exist.
C           - Wrong header written to xcd passing file. Replace IGRD
C             with JPRT in the header line.
C           - PRB output changed to write coefficient summed over
C             spin systems.
C           - Introduce b8corr to correct low Te problem
C             in recombination coefficients.
C           - Add source of ionisation data to comments with
C             call to b8winf. This also tidies up writing of the
C             comment block.
C
C VERSION:  1.10                                 DATE: 8/11/99
C MODIFIED: Martin O'Mullane
C           - Write effective zeros for SCD, MET and XCD data.
C - Removed NDMET from call to b8winf
C
C-----------------------------------------------------------------------
      CHARACTER*9         CPRTA(NDMET)
      CHARACTER*18        CSTRGA(NDLEV)
      CHARACTER*8         DATE
      CHARACTER*80        DSFULL,      DSNEXP,      DSNINC
      CHARACTER*3         TITLED
      CHARACTER*30        USER
      INTEGER             IA(NDLEV),   IBSELA(NDMET,NDMET),      IL
      INTEGER             ILA(NDLEV),  IMETR(NDMET)
      INTEGER             ISA(NDLEV),  IUNIT,       IUNT14,      IUNT15
      INTEGER             IUNT16,      IUNT17,      IUNT18,      IUNT19
      INTEGER             IUNT20,      IUNT21,      IUNT22,      IUNT23
      INTEGER             IZ,          IZ0,         IZ1,         MAXD
      INTEGER             MAXT,        NDDEN,       NDLEV,       NDMET
      INTEGER             NDTEM,       NMET,        NORD,        NPL
      INTEGER             NPL3,        NPLI,        NPLR
      LOGICAL             LHSEL,       LIOSEL,      LISEL,       LNORM
      LOGICAL             LNSEL,       LPSEL,       LRSEL
      LOGICAL             LSS04A(NDLEV,NDMET),      LSSETA(NDMET,NDMET)
      LOGICAL             LZSEL
      REAL*8              BWNOA(NDMET),             DENSA(NDDEN)
      REAL*8              FVCRED(NDMET,NDMET,NDTEM,NDDEN)
      REAL*8              FVCRPR(NDMET,NDMET,NDTEM,NDDEN)
      REAL*8              FVHRED(NDMET,NDMET,NDTEM,NDDEN)
      REAL*8              FVIONR(NDMET,NDMET,NDTEM,NDDEN)
      REAL*8              FVIRED(NDMET,NDMET,NDTEM,NDDEN)
      REAL*8              FVRRED(NDMET,NDMET,NDTEM,NDDEN)
      REAL*8              PH(NDTEM,NDDEN,NDMET)
      REAL*8              PL(NDMET,NDTEM,NDDEN)
```

```
REAL*8              PR(NDMET,NDTEM,NDDEN),     PRTWTA(NDMET)
REAL*8              PS(NDMET,NDMET,NDTEM,NDDEN)
REAL*8              RATMIA(NDDEN,NDMET),       RATPIA(NDDEN,NDMET)
REAL                STACK(NDLEV,NDMET,NDTEM,NDDEN)
REAL*8              STCKM(NDMET,NDTEM,NDDEN), SWVLN(NDMET)
REAL*8              TEVA(NDTEM), WA(NDLEV),    XJA(NDLEV)
```

3.64 b8wrps: Subroutine b8wrps from library adas2xx

```
      subroutine b8wrps( open15 , iunt15 , open13 , iunt13 ,
     &                   dsninc , dsfull , dsnexp , ibsela ,
     &                   titled , date   , user   ,
     &                   ndlev  , ndtem  , ndden  , ndmet  , ndtrn ,
     &                   lnorm  ,
     &                   iz     , iz0    , iz1    ,
     &                   il     , nmet   , nord   ,
     &                   maxt   , maxd   , icntr  , icnti  , icnth ,
     &                   isa    , ila    , xja    ,
     &                   cstrga , wa     ,
     &                   icnte  ,
     &                   ie1a   , ie2a   , aa     ,
     &                   imetr  , iordr  , teva   , densa  ,
     &                   npl    , nplr   , npli   , npl3   ,
     &                   lrsel  , lisel  , lhsel  , liosel ,
     &                   lpsel  , lzsel  , lnsel  ,
     &                   wvls   , wvll   , avlt   ,
     &                   stvr   , stvi   , stvh   ,
     &                   ratpia , ratmia , stack  ,
     &                   fvionr , sgrda  ,
     &                   lsseta , lss04a
     &                 )

C-----------------------------------------------------------------------
C
C  ******************** FORTRAN77 SUBROUTINE: B8WRPS ******************
C
C  PURPOSE:  To output data to pec and sxb passing files.
C
C  CALLING PROGRAM: ADAS208
C
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  IUNT15    = UNIT NUMBER FOR PECS.
C  INPUT : (L*4)  OPEN15    =.TRUE. IF SEC OUTPUT IS REQUIRED
C  INPUT : (I*4)  IUNT13    = UNIT NUMBER FOR SXBS.
C  INPUT : (L*4)  OPEN13    =.TRUE. IF SXB OUTPUT IS REQUIRED
C  INPUT : (C*44) DSNINC    = INPUT ADF04 DATA SET NAME
C  INPUT : (C*80) DSFULL    = INPUT ADF07 DATA SET NAME
C  INPUT : (C*80) DSNEXP    = INPUT EXPANSION FILE
C  INPUT : (I*4)  IBSELA(,) = IONISATION DATA BLOCK SELECTION INDICES
C                             1ST DIMENSION - (Z) ION METASTABLE COUNTER
C                             2ND DIMENSION - (Z+1) ION METASTABLE COUNTER
C  INPUT : (C*3)  TITLED    = ELEMENT SYMBOL.
C  INPUT : (C*8)  DATE      = CURRENT DATE.
C  INPUT : (C*30) USER      = FULL NAME OF AUTHOR.
C
C  INPUT : (I*4)  NDLEV     = MAXIMUM NUMBER OF LEVELS ALLOWED
C  INPUT : (I*4)  NDTEM     = MAXIMUM NUMBER OF TEMPERATURES ALLOWED
C  INPUT : (I*4)  NDDEN     = MAXIMUM NUMBER OF DENSITIES ALLOWED
C  INPUT : (I*4)  NDMET     = MAXIMUM NUMBER OF METASTABLES ALLOWED
C  INPUT : (I*4)  NDTRN     = PARAMETER = MAX. NO. OF TRANSITIONS ALLOWED
C  INPUT : (L*4)  LNORM     =.TRUE.  => IF NMET=1 THEN VARIOUS
C                                       EMISSIVITY OUTPUT FILES
C                                       NORMALISED TO STAGE TOT.POPULATN.
C                                       (** NORM TYPE = T)
C                            =.FALSE. => OTHERWISE NORMALISE TO IDENTIFIED
C                                       METASTABLE POPULATIONS.
```

```
C                                        (** NORM TYPE = M)
C  INPUT : (I*4)  IZ         =  RECOMBINED ION CHARGE READ
C  INPUT : (I*4)  IZ0        =         NUCLEAR CHARGE READ
C  INPUT : (I*4)  IZ1        = RECOMBINING ION CHARGE READ
C                               (NOTE: IZ1 SHOULD EQUAL IZ+1)
C
C  INPUT : (I*4)  IL         = NUMBER OF ENERGY LEVELS
C  INPUT : (I*4)  NMET       = NUMBER OF METASTABLES LEVELS: 1<=NMET<=NDMET
C  INPUT : (I*4)  NORD       = NUMBER OF ORDINARY LEVELS ('IL' - 'NMET')
C  INPUT : (I*4)  MAXT       = NUMBER OF INPUT TEMPERATURES ( 1 -> 'NDTEM')
C  INPUT : (I*4)  MAXD       = NUMBER OF INPUT DENSITIES ( 1 -> 'NDDEN')
C  INPUT : (I*4)  ICNTR      = NUMBER OF FREE ELECTRON RECOMBINATIONS INPUT
C  INPUT : (I*4)  ICNTI      = NUMBER OF LOWER STAGE IONISATIONS      INPUT
C  INPUT : (I*4)  ICNTH      = NO. OF CHARGE EXCHANGE RECOMBINATIONS INPUT
C  INPUT : (I*4)  ISA()      = MULTIPLICITY FOR LEVEL 'IA()'
C                               NOTE: (ISA-1)/2 = QUANTUM NUMBER (S)
C  INPUT : (I*4)  ILA()      = QUANTUM NUMBER (L) FOR LEVEL 'IA()'
C  INPUT : (R*8)  XJA()      = QUANTUM NUMBER (J-VALUE) FOR LEVEL 'IA()'
C                               NOTE: (2*XJA)+1 = STATISTICAL WEIGHT
C  INPUT : (C*18) CSTRGA()   = NOMENCLATURE/CONFIGURATION FOR LEVEL 'IA()'
C  INPUT : (R*8)  WA()       = ENERGY RELATIVE TO LEVEL 1 (CM-1)
C                               DIMENSION: LEVEL INDEX
C  INPUT : (I*4)  ICNTE      = NUMBER OF ELECTRON IMPACT TRANSITIONS INPUT
C  INPUT : (I*4)  IE1A()     = ELECTRON IMPACT TRANSITION:
C                                LOWER ENERGY LEVEL INDEX
C  INPUT : (I*4)  IE2A()     = ELECTRON IMPACT TRANSITION:
C                                UPPER ENERGY LEVEL INDEX
C  INPUT : (R*8)  AA()       = ELECTRON IMPACT TRANSITION: A-VALUE (SEC-1)
C  INPUT : (I*4)  IMETR()    = INDEX OF METASTABLE IN COMPLETE LEVEL LIST
C  INPUT : (I*4)  IORDR()    = INDEX OF ORDINARY LEVELS IN COMPLETE LEVEL
C                               LIST.
C  INPUT : (R*8)  TEVA()     = ELECTRON TEMPERATURES (UNITS: KELVIN)
C  INPUT : (R*8)  DENSA()    = ELECTRON DENSITIES  (UNITS: CM-3)
C  INPUT : (I*4)  NPL        = NO. OF METASTABLES OF(Z+1) ION ACCESSED
C                                 BY EXCITED STATE IONISATION IN COPASE
C                                 FILE WITH IONISATION POTENTIALS GIVEN
C                                 ON THE FIRST DATA LINE
C  INPUT : (I*4)  NPLR       = NO. OF ACTIVE METASTABLES OF (Z+1) ION
C  INPUT : (I*4)  NPLI       = NO. OF ACTIVE METASTABLES OF (Z+1) ION
C  INPUT : (I*4)  NPL3       = NO. OF ACTIVE METASTABLES OF (Z+1) ION
C  INPUT   (L*4)  LRSEL      = .TRUE.  => INCLUDE FREE ELECTRON
C                                         RECOMBINATION.
C                              = .FALSE. => DO NOT INCLUDE FREE ELECTRON
C                                         RECOMBINATION.
C  INPUT : (L*4)  LISEL      = .TRUE.  => INCLUDE ELECTRON IMPACT
C                                         IONISATION.
C                              = .FALSE. => DO NOT INCLUDE FREE ELECTRON
C                                         RECOMBINATION.
C  INPUT : (L*4)  LHSEL      = .TRUE.  => INCLUDE CHARGE TRANSFER FROM
C                                         NEUTRAL HYDROGREN.
C                              = .FALSE. => DO NOT INCLUDE CHARGE TRANSFER
C                                         FROM NEUTRAL HYDROGREN.
C  INPUT : (L*4)  LIOSEL     = .TRUE.  => INCLUDE IONISATION RATES
C                              = .FALSE. => DO NOT INCLUDE IONISATION RATES
C                                 FOR RECOM AND  3-BODY
C  INPUT : (L*4)  LPSEL      = .TRUE.  => INCLUDE PROTON COLLISIONS
C                              = .FALSE. =>DO NOT INCLUDE PROTON COLLISIONS
C  INPUT : (L*4)  LZSEL      = .TRUE.  => SCALE PROTON COLLISIONS WITH
C                                         PLASMA Z EFFECTIVE'ZEFF'.
C                              = .FALSE. => DO NOT SCALE PROTON COLLISIONS
C                                          WITH PLASMA Z EFFECTIVE 'ZEFF'.
```

```
C                                (ONLY USED IF 'LPSEL=.TRUE.')
C  INPUT : (L*4)  LNSEL     = .TRUE.  => INCLUDE PROJECTED BUNDLE-N DATA
C                                       FROM DATAFILE IF AVAILABLE
C                            = .FALSE. => DO NOT INCLUDE PROJECTED
C                                       BUNDLE-N DATA
C  INPUT : (R*8)  WVLS      = SHORT WAVELENGTH LIMIT FOR PEC & SXB (A)
C  INPUT : (R*8)  WVLL      = LONG WAVELENGTH LIMIT FOR PEC & SXB  (A)
C  INPUT : (R*8)  AVLT      = LOWER LIMIT OF A-VALUES FOR PEC & SXB
C
C  INPUT : (R*4) STVR(,,,)  = FREE ELECTRON RECOMBINATION COEFFICIENTS
C                              1st DIMENSION: ORDINARY LEVEL INDEX
C                              2nd DIMENSION: TEMPERATURE INDEX
C                              3rd DIMENSION: DENSITY INDEX
C                              4TH DIMENSION: PARENT INDEX
C  INPUT : (R*4) STVI(,,,)  = ELECTRON IMPACT IONISATION  COEFFICIENTS
C                              1st DIMENSION: ORDINARY LEVEL INDEX
C                              2nd DIMENSION: TEMPERATURE INDEX
C                              3rd DIMENSION: DENSITY INDEX
C                              4TH DIMENSION: PARENT INDEX
C  INPUT : (R*4) STVH(,,,)  =  CHARGE EXCHANGE COEFFICIENTS
C                              1st DIMENSION: ORDINARY LEVEL INDEX
C                              2nd DIMENSION: TEMPERATURE INDEX
C                              3rd DIMENSION: DENSITY INDEX
C  INPUT : (R*8) RATPIA(,)  = RATIO ( N(Z+1)/N(Z)  STAGE ABUNDANCIES )
C                              1ST DIMENSION: TEMP/DENS INDEX
C                              2ND DIMENSION: PARENT INDEX
C  INPUT : (R*8) RATMIA(,)  = RATIO ( N(Z-1)/N(Z)  STAGE ABUNDANCIES )
C                              1ST DIMENSION: TEMP/DENS INDEX
C                              2ND DIMENSION: PARENT INDEX
C  INPUT : (R*4) STACK(,,,) = POPULATION DEPENDENCE
C                              1st DIMENSION: ORDINARY LEVEL INDEX
C                              2nd DIMENSION: METASTABLE INDEX
C                              3rd DIMENSION: TEMPERATURE INDEX
C                              4th DIMENSION: DENSITY INDEX
C  INPUT : (R*8) FVIONR(,,,)= GEN. COLL. RAD. IONIS. RATE COEFFTS.
C                             1ST DIMENSION: (Z) ION METASTABLE INDEX
C                             2ND DIMENSION: (Z+1) ION METASTABLE INDEX
C                             3rd DIMENSION: TEMPERATURE INDEX
C                             4th DIMENSION: DENSITY INDEX
C  INPUT : (R*8)  SGRDA(,,) = GROUND & METASTABLE IONISATION RATE
C                             COEFFICIENTS  FROM SZD FILES (CM3 SEC-1)
C                             1ST DIMENSION: TEMPERATURE INDEX
C                             2ND DIMENSION: (Z) ION METASTABLE INDEX
C                             3RD DIMENSION: (Z+1) ION METASTABLE INDEX
C  INPUT : (L*4) LSSETA(,)  = .TRUE.  - MET. IONIS RATE SET IN B8GETS
C                             .FALSE.- MET. IONIS RATE NOT SET IN B8GETS
C                              1ST DIMENSION: (Z) ION METASTABLE INDEX
C                              2ND DIMENSION: (Z+1) ION METASTABLE INDEX
C  INPUT : (L*4) LSS04A(,)  = .TRUE. => IONIS. RATE SET IN ADF04 FILE:
C                             .FALSE.=> NOT SET IN ADF04 FILE
C                               1ST DIM: LEVEL INDEX
C                               2ND DIM: PARENT METASTABLE INDEX
C
C          (I*4) NOTRN      = PARAMETER = MAXIMUM NUMBER OF TRANSITIONS
C          (I*4) NDPEC      = PARAMETER = MAXIMUM NUMBER OF PECS PER
C                                         METASTABLE FOR OUTPUT
C          (I*4) METCNT     = COUNTER OF PECS FOR EACH METASTABLE
C
C          (I*4) I4UNIT     = FUNCTION (SEE ROUTINE SELECTION BELOW)
C
C          (I*4) I          = GENERAL USE
```

```
C          (I*4) IP         = GENERAL USE
C          (I*4) J          = GENERAL USE
C          (I*4) K          = GENERAL USE
C          (I*4) L          = GENERAL USE
C
C          (R*8) DUM1       = GENERAL USE- DUMMY
C          (R*8) DUM2       = GENERAL USE- DUMMY
C          (R*8) DUM3       = GENERAL USE- DUMMY
C
C ROUTINES:
C          ------------------------------------------------------------
C          i4unit     ADAS      Fetch unit number for output of messages
C          b8norm     ADAS      Perform  stage population normalisation
C          b8corp     ADAS      'fixes' low te problem in rec. data of pecs
C          b8winf     ADAS      Determines ionis. source and writes
C                               comment block
C          xxeiam     ADAS      Get 2-character name of element.
C          xxslen     ADAS      Returns lenght of string.
C          xxordr     ADAS      Sorts a real*8 array.
C
C-----------------------------------------------------------------------
C
C NOTES: Based on b8wr11, b8wr12 and hapecf.
C
C
C VERSION  : 1.1
C DATE     : 15-02-2006
C MODIFIED : Martin O'Mullane
C              - First version.
C
C VERSION  : 1.2
C DATE     : 08-03-2006
C MODIFIED : Martin O'Mullane
C              - Make sure that CX pec data which is 0.0 is written
C                as 1.0E-70 to avoid splining problems when reading
C                adf15 datasets.
C
C-----------------------------------------------------------------------
      CHARACTER*18        CSTRGA(NDLEV)
      CHARACTER*8         DATE
      CHARACTER*80        DSFULL,      DSNEXP,      DSNINC
      CHARACTER*3         TITLED
      CHARACTER*30        USER
      INTEGER             IBSELA(NDMET,NDMET),      ICNTE,       ICNTH
      INTEGER             ICNTI,       ICNTR,       IE1A(NDTRN)
      INTEGER             IE2A(NDTRN), IL,          ILA(NDLEV)
      INTEGER             IMETR(NDMET),             IORDR(NDLEV)
      INTEGER             ISA(NDLEV),  IUNT13,      IUNT15,      IZ
      INTEGER             IZ0,         IZ1,         MAXD,        MAXT
      INTEGER             NDDEN,       NDLEV,       NDMET,       NDTEM
      INTEGER             NDTRN,       NMET,        NORD,        NPL
      INTEGER             NPL3,        NPLI,        NPLR
      LOGICAL             LHSEL,       LIOSEL,      LISEL,       LNORM
      LOGICAL             LNSEL,       LPSEL,       LRSEL
      LOGICAL             LSS04A(NDLEV,NDMET),      LSSETA(NDMET,NDMET)
      LOGICAL             LZSEL,       OPEN13,      OPEN15
      REAL*8              AA(NDTRN),   AVLT,        DENSA(NDDEN)
      REAL*8              FVIONR(NDMET,NDMET,NDTEM,NDDEN)
      REAL*8              RATMIA(NDDEN,NDMET),      RATPIA(NDDEN,NDMET)
      REAL*8              SGRDA(NDTEM,NDMET,NDMET)
      REAL                STACK(NDLEV,NDMET,NDTEM,NDDEN)
```

```
REAL                STVH(NDLEV,NDTEM,NDDEN,NDMET)
REAL                STVI(NDLEV,NDTEM,NDDEN,NDMET)
REAL                STVR(NDLEV,NDTEM,NDDEN,NDMET)
REAL*8              TEVA(NDTEM), WA(NDLEV),   WVLL,        WVLS
REAL*8              XJA(NDLEV)
```

3.65 b9data: Subroutine b9data from library adas2xx

```
       SUBROUTINE B9DATA( IUNIT  , NDLEV  , NDTRN , NDMET ,
     &                    TITLED , IZ     , IZ0   , IZ1   , BWNO  ,
     &                    NPL    , BWNOA  , LBSETA, PRTWTA, CPRTA ,
     &                    IL     ,
     &                    IA     , CSTRGA , ISA   , ILA   , XJA   , WA ,
     &                    CPLA   , NPLA   , IPLA  , ZPLA  ,
     &                    NV     , SCEF   ,
     &                    ITRAN  , MAXLEV ,
     &                    TCODE  , I1A    , I2A   , AVAL  , SCOM  , ITYP
     &                  )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: B9DATA *********************
C
C  PURPOSE:  TO FETCH DATA FROM INPUT COPASE DATA SET, INCLUDING
C            MULTIPLE PARENTS ON FREE-ELECTRON AND CHARGE EXCHANGE
C            ON RECOMBINATION, INCLUSION OF EXPLICIT CONTRIBUTIONS BY
C            IONISATION.
C
C            IMPROVEMENT OF AUTOMATIC IONISATION CALC. BY INCLUDING
C            ASSIGNMENT OF FINAL STATE PARENT.
C
C  CALLING PROGRAM: ADAS209
C
C  DATA:
C           THE 'REAL' DATA IN THE FILE IS REPRESENTED IN AN ABBREVIATED
C           FORM WHICH OMITS THE "D" OR "E" EXPONENT SPECIFIER.
C           e.g. 1.23D-06 or 1.23E-06 IS REPRESENTED AS 1.23-06
C                6.75D+07 or 6.75E+07 IS REPRESENTED AS 6.75+07
C
C           THEREFORE THE FORM OF EACH 'REAL' NUMBER IN THE DATA SET IS:
C                          N.NN+NN or N.NN-NN
C
C           THE UNITS USED IN THE DATA FILE ARE TAKEN AS FOLLOWS:
C
C           IONISATION POTENTIAL: WAVE NUMBER (CM-1)
C           INDEX LEVEL ENERGIES: WAVE NUMBER (CM-1)
C           TEMPERATURES        : KELVIN
C           A-VALUES            : SEC-1
C           GAMMA-VALUES        :
C           RATE COEFFT.        : CM3 SEC-1
C
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  IUNIT   = UNIT TO WHICH INPUT FILE IS ALLOCATED
C  INPUT : (I*4)  NDLEV   = MAXIMUM NUMBER OF LEVELS THAT CAN BE READ
C  INPUT : (I*4)  NDTRN   = MAX. NUMBER OF TRANSITIONS THAT CAN BE READ
C  INPUT : (I*4)  NDMET   = MAX. NUMBER OF METASTABLES ALLOWED
C
C  OUTPUT: (C*3)  TITLED  = ELEMENT SYMBOL.
C  OUTPUT: (I*4)  IZ      =  RECOMBINED ION CHARGE READ
C  OUTPUT: (I*4)  IZ0     =         NUCLEAR CHARGE READ
C  OUTPUT: (I*4)  IZ1     = RECOMBINING ION CHARGE READ
C                           (NOTE: IZ1 SHOULD EQUAL IZ+1)
C  OUTPUT: (R*8)  BWNO    = IONISATION POTENTIAL (CM-1) OF LOWEST PARENT
C  OUTPUT: (I*4)  NPL     = NUMBER OF PARENTS ON FIRST LINE AND USED
C                           IN LEVEL ASSIGNMENTS
C  OUTPUT: (R*8)  BWNOA() = IONISATION POTENTIAL (CM-1) OF PARENTS
```

```
C  OUTPUT: (L*4)  LBSETA()= .TRUE.  - PARENT WEIGHT SET FOR BWNOA()
C                           .FALSE. - PARENT WEIGHT NOT SET FOR BWNOA()
C  OUTPUT: (R*8)  PRTWTA()= PARENT WEIGHT FOR BWNOA()
C  OUTPUT: (C*9)  CPRTA() = PARENT NAME IN BRACKETS
C
C  OUTPUT: (I*4)  IL      = INPUT DATA FILE: NUMBER OF ENERGY LEVELS
C
C  OUTPUT: (I*4)  IA()    = ENERGY LEVEL INDEX NUMBER
C  OUTPUT: (C*18) CSTRGA()= NOMENCLATURE/CONFIGURATION FOR LEVEL 'IA()'
C  OUTPUT: (I*4)  ISA()   = MULTIPLICITY FOR LEVEL 'IA()'
C                           NOTE: (ISA-1)/2 = QUANTUM NUMBER (S)
C  OUTPUT: (I*4)  ILA()   = QUANTUM NUMBER (L) FOR LEVEL 'IA()'
C  OUTPUT: (R*8)  XJA()   = QUANTUM NUMBER (J-VALUE) FOR LEVEL 'IA()'
C                           NOTE: (2*XJA)+1 = STATISTICAL WEIGHT
C  OUTPUT: (R*8)  WA()    = ENERGY RELATIVE TO LEVEL 1 (CM-1) FOR LEVEL
C                           'IA()'
C  OUTPUT: (C*1)  CPLA()  = CHAR. SPECIFYING 1ST PARENT FOR LEVEL 'IA()'
C                                INTEGER - PARENT IN BWNOA() LIST
C                                'BLANK' - PARENT BWNOA(1)
C                                  'X'   - DO NOT ASSIGN A PARENT
C  OUTPUT: (I*4)  NPLA()  = NO. OF PARENT/ZETA CONTRIBUTIONS TO IONIS.
C                           OF LEVEL
C  OUTPUT: (I*4)  IPLA(,) = PARENT INDEX FOR CONTRIBUTIONS TO IONIS.
C                           OF LEVEL
C                           1ST DIMENSION: PARENT INDEX
C                           2ND DIMENSION: LEVEL INDEX
C  OUTPUT: (I*4)  ZPLA(,) = EFF. ZETA PARAM. FOR CONTRIBUTIONS TO IONIS.
C                           OF LEVEL
C                           1ST DIMENSION: PARENT INDEX
C                           2ND DIMENSION: LEVEL INDEX
C
C  OUTPUT: (I*4)  NV      = INPUT DATA FILE: NUMBER OF GAMMA/TEMPERATURE
C                           PAIRS FOR A GIVEN TRANSITION.
C  OUTPUT: (R*8)  SCEF()  = INPUT DATA FILE: ELECTRON TEMPERATURES (K)
C                           (INITIALLY JUST THE MANTISSA. SEE 'ITPOW()')
C                           (NOTE: TE=TP=TH IS ASSUMED)
C
C  OUTPUT: (I*4)  ITRAN   = INPUT DATA FILE: NUMBER OF TRANSITIONS
C  OUTPUT: (I*4)  MAXLEV  = HIGHEST INDEX LEVEL IN READ TRANSITIONS
C
C  OUTPUT: (C*1)  TCODE() = TRANSITION: DATA TYPE POINTER:
C                           ' ' => Electron Impact   Transition
C                           'P' => Proton   Impact   Transition
C                           'H' => Charge   Exchange Recombination
C                           'R' => Free     Electron Recombination
C                           'I' => Coll. ionisation from lower stage ion
C  OUTPUT: (I*4)  I1A()   = TRANSITION:
C                            LOWER ENERGY LEVEL INDEX (CASE ' ' & 'P')
C                            SIGNED PARENT NDEX (CASE 'H','R' & 'I')
C  OUTPUT: (I*4)  I2A()   = TRANSITION:
C                            UPPER ENERGY LEVEL INDEX (CASE ' ' & 'P')
C                            CAPTURING LEVEL INDEX (CASE 'H','R' & 'I')
C  OUTPUT: (R*8)  AVAL()  = TRANSITION:
C                            A-VALUE (SEC-1)          (CASE ' ')
C                            NEUTRAL BEAM ENERGY      (CASE 'H')
C                            NOT USED             (CASE 'P','R' & 'I')
C  OUTPUT: (R*8)  SCOM(,) = TRANSITION:
C                            GAMMA VALUES             (CASE ' ' & 'P')
C                            RATE COEFFT.(CM3 SEC-1)(CASE 'H','R' & 'I')
C                           1ST DIMENSION - TEMPERATURE 'SCEF()'
C                           2ND DIMENSION - TRANSITION NUMBER
```

```
C
C          (I*4)  NVMAX   = PARAMETER = MAX. NUMBER OF TEMPERATURES
C                                       THAT CAN BE READ IN.
C          (I*4)  MTIED   = PARAMETER = MUST BE GREATER THAN OR EQUAL TO
C                                       THE MAX. NO. OF LEVELS.
C          (R*8)  DZERO   = PARAMETER = MINIMUM VALUE FOR 'AVAL()' AND
C                                       'SCOM()' ARRAYS = 1.0D-30
C
C          (I*4)  I4UNIT  = FUNCTION (SEE ROUTINE SELECTION BELOW)
C          (I*4)  IQS     = X-SECT DATA FORMAT SELECTOR
C                           NOTE: IQS=3 ONLY ALLOWED IN THIS PROGRAM
C          (I*4)  IFAIL   = FAILURE NUMBER FROM B9PARS AND B9PRS1
C          (I*4)  I       = GENERAL USE.
C          (I*4)  IABT    = RETURN CODE FROM 'R(FCTN' (0 => NO ERROR)
C                           OR FROM INTERROGATION OF 'C7'
C          (I*4)  J       = GENERAL USE.
C          (I*4)  J1      = INPUT DATA FILE - SELECTED TRANSITION:
C                            LOWER ENERGY LEVEL INDEX (CASE ' ' & 'P')
C          (I*4)  J2      = INPUT DATA FILE - SELECTED TRANSITION:
C                            UPPER ENERGY LEVEL INDEX (CASE ' ' & 'P')
C                            CAPTURING    LEVEL INDEX (CASE 'H' & 'R')
C          (I*4)  LENCST  = BYTE LENGTH OF STRING CSTRGA()
C          (I*4)  ILINE   = ENERGY LEVEL INDEX FOR CURRENT LINE
C          (I*4)  IRECL   = RECORD LENGTH OF INPUT DATASET (<=128)
C          (I*4)  IAPOW   = EXPONENT OF 'AVALM'
C          (I*4)  IGPOW() = EXPONENT OF 'GAMMA()'
C          (I*4)  ITPOW() = TEMPERATURES - EXPONENT
C                           NOTE: MANTISSA INITIALLY KEPT IN 'SCEF()'
C
C          (R*4)  ZF      = SHOULD BE EQUIVALENT TO 'IZ1'
C
C          (R*8)  AVALM   = INPUT DATA FILE - SELECTED TRANSITION:
C                           MANTISSA OF:   ('IAPOW' => EXPONENT)
C                            A-VALUE (SEC-1)          (CASE ' ')
C                            NEUTRAL BEAM ENERGY      (CASE 'H')
C                            NOT USED              (CASE 'P','R' & 'I')
C          (R*8)  GAMMA() = INPUT DATA FILE - SELECTED TRANSITION:
C                           MANTISSA OF: ('IGPOW()' => EXPONENT)
C                            GAMMA VALUES             (CASE ' ' & 'P')
C                            RATE COEFFT.(CM3 SEC-1)(CASE 'H','R' & 'I')
C                           DIMENSION => TEMPERATURE 'SCEF()'
C
C          (C*7)  C7      = USED TO PARSE VALUE FOR XJA()
C          (C*7)  CDELIM  = DELIMITERS FOR INPUT OF DATA FROM HEADERS
C          (C*18) C18     = USED TO PARSE VALUE TO CSTRGA()
C          (C*18) C18T    = COPY OF C18 : UNSATISFACTORY METHOD OF
C                           AVOIDING COMPILER REFERENCE ERROR :
C                           DHB 07.04.95
C          (C*80) CLINE   = CURRENT ENERGY LEVEL INDEX PARAMETER LINE
C          (C*75) STRING  = TAIL STRING OF 1ST DATA LINE FOR PARSING
C          (C*56) STRG1   = TAIL STRING OF LEVEL SPEC LINES FOR PARSING
C          (C*128)BUFFER  = GENERAL STRING BUFFER STORAGE
C          (C*3)  CITPOW()= USED TO PARSE VALUES TO ITPOW()
C          (C*5)  CSCEF() = USED TO PARSE VALUES TO SCEF()
C
C          (L*4)  LDATA   = IDENTIFIES  WHETHER  THE END OF AN  INPUT
C                           SECTION IN THE DATA SET HAS BEEN LOCATED.
C                           (.TRUE. => END OF SECTION REACHED)
C          (L*4)  LTCHR   = .TRUE.  => CURRENT 'TCODE()' = 'H' OR 'R'
C                                                            OR 'I'
C                         = .FALSE. => CURRENT 'TCODE()'.NE.'H' OR 'R'
```

```
C                                                                 OR 'I'
C          (L*4)  LTCPR   = .TRUE.  => CURRENT 'TCODE()' = 'P' OR 'R'
C                                                               OR 'I'
C                         = .FALSE. => CURRENT 'TCODE()'.NE. 'P' OR 'R'
C                                                                OR 'I'
C          (L*4)  LERROR  = .TRUE.  => UNTIED LEVEL FOUND
C                         = .FALSE. => ALL LEVELS TIED
C          (L*4)  LTIED() = .TRUE.  => SPECIFIED LEVEL TIED
C                         = .FALSE. => SPECIFIED LEVEL IS UNTIED
C                           DIMENSION => LEVEL INDEX
C OUTPUT:  (I*4)  ITYP    = RESOLUTION OF PARENT METASTABLES
C                           1 - LS RESOLVED
C                           2 - LSJ RESOLVED
C                           3 - UNIDENTIFIED
C
C
C NOTE:            LTCHR        LTCPR         TCODE()
C                 ---------------------------------
C                 .TRUE.       .TRUE.    =>     'R','I'
C                 .TRUE.       .FALSE.   =>     'H'
C                 .FALSE.      .TRUE.    =>     'P'
C                 .FALSE.      .FALSE.   =>     ' '
C
C        FOR A-VALUES & GAMMA-VALUES ENTRIES LESS THAN 'DZERO' ARE TAKEN
C        AS BEING EQUAL TO DZERO. THIS AFFECTS THE 'AVAL()' AND 'SCOM()'
C        ARRAYS.
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4UNIT     ADAS      FETCH UNIT NUMBER FOR OUTPUT OF MESSAGES
C          R8FCTN     ADAS      CONVERTS FROM CHARACTER TO REAL VARIABLE
C          I4FCTN     ADAS      CONVERTS FROM CHAR. TO INTEGER  VARIABLE
C          XXWORD     ADAS      PARSES A STRING INTO SEPARATE WORDS
C                               FOR ' ()<>{}' DELIMITERS
C
C AUTHOR:  HP SUMMERS   (REVISION OF BXDATA BY PE BRIDEN)
C          K1/1/57
C          JET EXT. 4941
C
C DATE:    11/06/92
C
C UPDATE:   9/07/93  HPS - USE NEW VERSIONS OF PARSING ROUTINES
C                          B8PARS AND B8PRS1
C UPDATE:  12/07/93  HPS - REVISE TO CONSISTENCY WITH  BXDATA
C                          AT 25/07/93.
C UPDATE:  11/05/95  HPS - ADDED CPRTA TO PARAMETER LIST.ALTERED
C                          'READ()BUFFER' TO BE CONSISTENT WITH IDL-ADAS
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 27-06-95
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - PUT UNDER SCCS CONTROL
C
C VERSION: 1.2                          DATE: 19-01-96
C MODIFIED: DAVID BROOKS (UNIVERSITY OF STRATHCLYDE)/TIM HAMMOND
C               - INCREASED LENGTH OF CPRTA FROM 4 TO 9 &
C                 STRING FROM 55 TO 75 IN LINE WITH
C                 MODIFICATIONS TO ACCOMODATE J-RESOLVED
C                 PARENT METASTABLES IN THE DATASETS.
```

```
C               - INCREASED LENGTH OF CLINE TO 92 & STRG1 TO
C                 56. ALTERED FORMAT NO. 1003 & READING OF
C                 CLINE FORMAT TO ACCOMMODATE CHANGES.
C
C VERSION: 1.3                          DATE: 26-01-96
C MODIFIED: DAVID BROOKS
C               - PASSED ITYP THROUGH TO MAIN PROGRAM
C
C
C VERSION: 1.4 DATE: 18/04/96
C UPDATE:  WILLIAM OSBORN
C - INCREASED MTIED TO SAME AS NDLEV
C
C VERSION: 1.5 DATE: 18/11/98
C UPDATE:  DAVID BROOKS
C - ALLOWED LEVELS TO 250.
C
C VERSION:  1.6                         DATE: 01/11/2002
C MODIFIED: Martin O'Mullane
C           - Can handle S lines correctly.
C
C UPDATE:   1.5                         DATE: 17/05/07
C MODIFIED: Allan Whiteford
C           - Updated comments as part of subroutine documentation
C             procedure.
C
C-----------------------------------------------------------------------
      CHARACTER           CPLA(NDLEV)
      CHARACTER*9         CPRTA(NDMET)
      CHARACTER*(*)       CSTRGA(NDLEV)
      CHARACTER           TCODE(NDTRN)
      CHARACTER*3         TITLED
      INTEGER             I1A(NDTRN),  I2A(NDTRN),  IA(NDLEV),   IL
      INTEGER             ILA(NDLEV),  IPLA(NDMET,NDLEV)
      INTEGER             ISA(NDLEV),  ITRAN,       ITYP,        IUNIT
      INTEGER             IZ,          IZ0,         IZ1,         MAXLEV
      INTEGER             NDLEV,       NDMET,       NDTRN,       NPL
      INTEGER             NPLA(NDLEV), NV
      LOGICAL             LBSETA(NDMET)
      REAL*8              AVAL(NDTRN), BWNO,        BWNOA(NDMET)
      REAL*8              PRTWTA(NDMET),            SCEF(NVMAX)
      REAL*8              SCOM(NVMAX,NDTRN),        WA(NDLEV)
      REAL*8              XJA(NDLEV),  ZPLA(NDMET,NDLEV)
```

3.66 b9pars: Subroutine b9pars from library adas2xx

```
       SUBROUTINE B9PARS(NDMET,STRING,NPT,BWNOA,LSETA,
     &                   PRTWTA,CPRTA,IFAIL,ITYPE)
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: B9PARS *********************
C
C  PURPOSE:  TO ANALYSE THE TAIL CHARACTER STRING OF THE FIRST LINE OF
C            A SPECIFIC ION FILE INTO BINDING WAVE NUMBERS FOR DIFFERENT
C            PARENTS AND STATISTICAL WEIGHTS FOR THE PARENTS.
C            MODIFICATION OF B8PARS.
C
C  CALLING PROGRAM: ADAS209
C
C  NOTES: DETECT  -  BINDING WAVE NUMBER WHICH PRECEED TERM ASSIGNATION
C                 -   TERM CONTAINED IN '(..)'.
C
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  NDMET    =  MAXIMUM NUMBER OF METASTABLES ALLOWED
C  INPUT : (C*(*))STRING   =  STRING TO BE PARSED
C
C  OUTPUT: (I*4)  NPT      =  NUMBER OF BINDING WAVE NUMBERS DETECTED
C  OUTPUT: (L*4)  LSETA()  =  .TRUE.  - PARENT TERM SET FOR THIS W.NO.
C                             .FALSE. - PARENT TERM NOT SET FOR W.NO.
C  OUTPUT: (L*4)  LFND     =  .TRUE.  - L QUANTUM NUMBER PRESENT IN
C                                       STRING
C                             .FALSE. - NO L QUANTUM NUMBER DETECTED
C  OUTPUT: (R*8)  BWNOA()  =  BINDING WAVE NUMBERS
C  OUTPUT: (R*8)  PRTWTA() =  PARENT STATISTICAL WEIGHTS
C  OUTPUT: (C*9)  CPRTA()  =  PARENT NAME IN BRACKETS
C  OUTPUT: (I*4)  IFAIL    =  0 - SUBROUTINE CONCLUDES CORRECTLY
C                             1 - FAULT DETECTED IN SUBROUTINE
C                             2 - SINGLE IONISATION POTENTIAL DETECTED
C
C          (I*4)  MAXWRD   =  MAXIMUM NUMBER OF WORDS SOUGHT INITIALLY
C                             INITIALLY, FINALLY NUMBER ACTUALLY FOUND
C          (I*4)  NFIRST   =  FIRST WORD TO BE EXTRACTED FROM STRING
C          (I*4)  IFIRST() =  INDEX OF FIRST CHAR. OF WORD () IN STRING
C          (I*4)  ILAST()  =  INDEX OF LAST  CHAR. OF WORD () IN STRING
C          (I*4)  IWORDS   =  NUMBER OF WORDS FOUND IN STRING
C          (I*4)  IABT     =  FAILURE NUMBER FROM R8FCTN
C          (I*4)  NCHAR    =  NUMBER OF CHARACTERS IN SUBSTRING
C          (I*4)  I        =  GENERAL USE
C          (I*4)  J        =  GENERAL USE
C          (I*4)  K        =  GENERAL USE
C          (I*4)  IC       =  GENERAL USE
C OUTPUT:  (I*4)  ITYPE    =  RESOLUTION OF PARENT METASTABLES
C                             1 - LS RESOLVED
C                             2 - LSJ RESOLVED
C                             3 - ARBITRARY RESOLUTION
C          (I*4)  ITP      =  FLAG FOR INCOMPATIBLE TYPES
C          (I*4)  ITYP     =  COPY OF CURRENT ITYPE
C          (I*4)  KMRK     =  POSITION MARKER IN THE STRING FOR PARENT
C                             L QUANTUM NUMBER
C          (R*8)  TWTA()   =  (2L+1) VALUE FOR PARENT L QUANTUM NUMBER
C          (C*1)  CTRMA()  =  PARENT L QUANTUM NUMBER LETTER SET
C
```

```
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4UNIT     ADAS      FETCH UNIT NUMBER FOR OUTPUT OF MESSAGES
C          R8FCTN     ADAS      CONVERTS FROM CHARACTER TO REAL VARIABLE
C          I4FCTN     ADAS      CONVERTS FROM CHAR. TO INTEGER  VARIABLE
C          XXWORD     ADAS      PARSES A STRING INTO SEPARATE WORDS
C                               FOR ' ()<>{}' DELIMITERS
C
C AUTHOR:  HP SUMMERS
C          K1/1/57
C          JET EXT. 4941
C
C DATE:    22/06/92
C
C UPDATE:   8/07/93 - HPS  ALTERED TO USE XXWORD PARSING ROUTINE
C
C UPDATE:  11/05/95 - HPS  ADD CPRTA TO PARAMETER LIST
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 27-06-95
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - PUT UNDER SCCS CONTROL
C
C VERSION: 1.2                          DATE: 13/11/95
C MODIFIED: DAVID BROOKS (UNIVERSITY OF STRATHCLYDE)
C               - EXTENDED ROUTINE TO HANDLE J/ARBITRARY
C                 RESOLVED PARENT METASTABLE INFORMATION
C                 IN THE FIRST DATA CARD STRING. INTRODUCED
C                 ITYPE TO FLAG RESOLUTION IF REQUIRED.
C
C VERSION: 1.3                          DATE: 26-01-96
C MODIFIED: DAVID BROOKS
C               - PASSED ITYPE FLAG OUT.
C
C-----------------------------------------------------------------------
      CHARACTER*9         CPRTA(NDMET)
      CHARACTER*(*)       STRING
      INTEGER             IFAIL,       ITYPE,       NDMET,       NPT
      LOGICAL             LSETA(NDMET)
      REAL*8              BWNOA(NDMET),             PRTWTA(NDMET)
```

3.67 badata: Subroutine badata from library adas2xx

```
       SUBROUTINE BADATA( IUNIT  , NDLEV  , NDTRN , NDMET ,
     &                    TITLED , IZ     , IZ0   , IZ1   , BWNO  ,
     &                    NPL    , BWNOA  , LBSETA, PRTWTA, CPRTA ,
     &                    IL     ,
     &                    IA     , CSTRGA , ISA   , ILA   , XJA   , WA ,
     &                    CPLA   , NPLA   , IPLA  , ZPLA  ,
     &                    NV     , SCEF   ,
     &                    ITRAN  , MAXLEV ,
     &                    TCODE  , I1A    , I2A   , AVAL  , SCOM
     &                  )
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: BADATA *********************
C
C  PURPOSE:  TO FETCH DATA FROM INPUT COPASE DATA SET, INCLUDING
C            MULTIPLE PARENTS ON FREE-ELECTRON AND CHARGE EXCHANGE
C            ON RECOMBINATION, INCLUSION OF EXPLICIT CONTRIBUTIONS BY
C            IONISATION.
C
C            IMPROVEMENT OF AUTOMATIC IONISATION CALC. BY INCLUDING
C            ASSIGNMENT OF FINAL STATE PARENT.
C
C  CALLING PROGRAM: ADAS210
C
C  DATA:
C           THE 'REAL' DATA IN THE FILE IS REPRESENTED IN AN ABBREVIATED
C           FORM WHICH OMITS THE "D" OR "E" EXPONENT SPECIFIER.
C           e.g. 1.23D-06 or 1.23E-06 IS REPRESENTED AS 1.23-06
C                6.75D+07 or 6.75E+07 IS REPRESENTED AS 6.75+07
C
C           THEREFORE THE FORM OF EACH 'REAL' NUMBER IN THE DATA SET IS:
C                          N.NN+NN or N.NN-NN
C
C           THE UNITS USED IN THE DATA FILE ARE TAKEN AS FOLLOWS:
C
C           IONISATION POTENTIAL: WAVE NUMBER (CM-1)
C           INDEX LEVEL ENERGIES: WAVE NUMBER (CM-1)
C           TEMPERATURES        : KELVIN
C           A-VALUES            : SEC-1
C           GAMMA-VALUES        :
C           RATE COEFFT.        : CM3 SEC-1
C
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  IUNIT   = UNIT TO WHICH INPUT FILE IS ALLOCATED
C  INPUT : (I*4)  NDLEV   = MAXIMUM NUMBER OF LEVELS THAT CAN BE READ
C  INPUT : (I*4)  NDTRN   = MAX. NUMBER OF TRANSITIONS THAT CAN BE READ
C  INPUT : (I*4)  NDMET   = MAX. NUMBER OF METASTABLES ALLOWED
C
C  OUTPUT: (C*3)  TITLED  = ELEMENT SYMBOL.
C  OUTPUT: (I*4)  IZ      =  RECOMBINED ION CHARGE READ
C  OUTPUT: (I*4)  IZ0     =         NUCLEAR CHARGE READ
C  OUTPUT: (I*4)  IZ1     = RECOMBINING ION CHARGE READ
C                           (NOTE: IZ1 SHOULD EQUAL IZ+1)
C  OUTPUT: (R*8)  BWNO    = IONISATION POTENTIAL (CM-1) OF LOWEST PARENT
C  OUTPUT: (I*4)  NPL     = NUMBER OF PARENTS ON FIRST LINE AND USED
C                           IN LEVEL ASSIGNMENTS
```

```
C  OUTPUT: (R*8)  BWNOA() = IONISATION POTENTIAL (CM-1) OF PARENTS
C  OUTPUT: (L*4)  LBSETA()= .TRUE.  - PARENT WEIGHT SET FOR BWNOA()
C                           .FALSE. - PARENT WEIGHT NOT SET FOR BWNOA()
C  OUTPUT: (R*8)  PRTWTA()= PARENT WEIGHT FOR BWNOA()
C  OUTPUT: (C*9)  CPRTA() = PARENT NAME IN BRACKETS
C
C  OUTPUT: (I*4)  IL      = INPUT DATA FILE: NUMBER OF ENERGY LEVELS
C
C  OUTPUT: (I*4)  IA()    = ENERGY LEVEL INDEX NUMBER
C  OUTPUT: (C*18) CSTRGA()= NOMENCLATURE/CONFIGURATION FOR LEVEL 'IA()'
C  OUTPUT: (I*4)  ISA()   = MULTIPLICITY FOR LEVEL 'IA()'
C                           NOTE: (ISA-1)/2 = QUANTUM NUMBER (S)
C  OUTPUT: (I*4)  ILA()   = QUANTUM NUMBER (L) FOR LEVEL 'IA()'
C  OUTPUT: (R*8)  XJA()   = QUANTUM NUMBER (J-VALUE) FOR LEVEL 'IA()'
C                           NOTE: (2*XJA)+1 = STATISTICAL WEIGHT
C  OUTPUT: (R*8)  WA()    = ENERGY RELATIVE TO LEVEL 1 (CM-1) FOR LEVEL
C                           'IA()'
C  OUTPUT: (C*1)  CPLA()  = CHAR. SPECIFYING 1ST PARENT FOR LEVEL 'IA()'
C                                INTEGER - PARENT IN BWNOA() LIST
C                                'BLANK' - PARENT BWNOA(1)
C                                  'X'   - DO NOT ASSIGN A PARENT
C  OUTPUT: (I*4)  NPLA()  = NO. OF PARENT/ZETA CONTRIBUTIONS TO IONIS.
C                           OF LEVEL
C  OUTPUT: (I*4)  IPLA(,) = PARENT INDEX FOR CONTRIBUTIONS TO IONIS.
C                           OF LEVEL
C                           1ST DIMENSION: PARENT INDEX
C                           2ND DIMENSION: LEVEL INDEX
C  OUTPUT: (I*4)  ZPLA(,) = EFF. ZETA PARAM. FOR CONTRIBUTIONS TO IONIS.
C                           OF LEVEL
C                           1ST DIMENSION: PARENT INDEX
C                           2ND DIMENSION: LEVEL INDEX
C
C  OUTPUT: (I*4)  NV      = INPUT DATA FILE: NUMBER OF GAMMA/TEMPERATURE
C                           PAIRS FOR A GIVEN TRANSITION.
C  OUTPUT: (R*8)  SCEF()  = INPUT DATA FILE: ELECTRON TEMPERATURES (K)
C                           (INITIALLY JUST THE MANTISSA. SEE 'ITPOW()')
C                           (NOTE: TE=TP=TH IS ASSUMED)
C
C  OUTPUT: (I*4)  ITRAN   = INPUT DATA FILE: NUMBER OF TRANSITIONS
C  OUTPUT: (I*4)  MAXLEV  = HIGHEST INDEX LEVEL IN READ TRANSITIONS
C
C  OUTPUT: (C*1)  TCODE() = TRANSITION: DATA TYPE POINTER:
C                           ' ' => Electron Impact   Transition
C                           'P' => Proton   Impact   Transition
C                           'H' => Charge   Exchange Recombination
C                           'R' => Free     Electron Recombination
C                           'I' => Coll. ionisation from lower stage ion
C  OUTPUT: (I*4)  I1A()   = TRANSITION:
C                            LOWER ENERGY LEVEL INDEX (CASE ' ' & 'P')
C                            SIGNED PARENT NDEX (CASE 'H','R' & 'I')
C  OUTPUT: (I*4)  I2A()   = TRANSITION:
C                            UPPER ENERGY LEVEL INDEX (CASE ' ' & 'P')
C                            CAPTURING LEVEL INDEX (CASE 'H','R' & 'I')
C  OUTPUT: (R*8)  AVAL()  = TRANSITION:
C                            A-VALUE (SEC-1)          (CASE ' ')
C                            NEUTRAL BEAM ENERGY      (CASE 'H')
C                            NOT USED             (CASE 'P','R' & 'I')
C  OUTPUT: (R*8)  SCOM(,) = TRANSITION:
C                            GAMMA VALUES             (CASE ' ' & 'P')
C                            RATE COEFFT.(CM3 SEC-1)(CASE 'H','R' & 'I')
C                           1ST DIMENSION - TEMPERATURE 'SCEF()'
```

```
C                          2ND DIMENSION - TRANSITION NUMBER
C
C          (I*4)  NVMAX   = PARAMETER = MAX. NUMBER OF TEMPERATURES
C                                       THAT CAN BE READ IN.
C          (I*4)  MTIED   = PARAMETER = MUST BE GREATER THAN OR EQUAL TO
C                                       THE MAX. NO. OF LEVELS.
C          (R*8)  DZERO   = PARAMETER = MINIMUM VALUE FOR 'AVAL()' AND
C                                       'SCOM()' ARRAYS = 1.0D-30
C
C          (I*4)  I4UNIT  = FUNCTION (SEE ROUTINE SELECTION BELOW)
C          (I*4)  IQS     = X-SECT DATA FORMAT SELECTOR
C                           NOTE: IQS=3 ONLY ALLOWED IN THIS PROGRAM
C          (I*4)  IFAIL   = FAILURE NUMBER FROM B9PARS AND B9PRS1
C          (I*4)  I       = GENERAL USE.
C          (I*4)  IABT    = RETURN CODE FROM 'R(FCTN' (0 => NO ERROR)
C                           OR FROM INTERROGATION OF 'C7'
C          (I*4)  J       = GENERAL USE.
C          (I*4)  J1      = INPUT DATA FILE - SELECTED TRANSITION:
C                            LOWER ENERGY LEVEL INDEX (CASE ' ' & 'P')
C          (I*4)  J2      = INPUT DATA FILE - SELECTED TRANSITION:
C                            UPPER ENERGY LEVEL INDEX (CASE ' ' & 'P')
C                            CAPTURING    LEVEL INDEX (CASE 'H' & 'R')
C          (I*4)  LENCST  = BYTE LENGTH OF STRING CSTRGA()
C          (I*4)  ILINE   = ENERGY LEVEL INDEX FOR CURRENT LINE
C          (I*4)  IRECL   = RECORD LENGTH OF INPUT DATASET (<=128)
C          (I*4)  IAPOW   = EXPONENT OF 'AVALM'
C          (I*4)  IGPOW() = EXPONENT OF 'GAMMA()'
C          (I*4)  ITPOW() = TEMPERATURES - EXPONENT
C                           NOTE: MANTISSA INITIALLY KEPT IN 'SCEF()'
C
C          (R*4)  ZF      = SHOULD BE EQUIVALENT TO 'IZ1'
C
C          (R*8)  AVALM   = INPUT DATA FILE - SELECTED TRANSITION:
C                           MANTISSA OF:   ('IAPOW' => EXPONENT)
C                            A-VALUE (SEC-1)          (CASE ' ')
C                            NEUTRAL BEAM ENERGY      (CASE 'H')
C                            NOT USED              (CASE 'P','R' & 'I')
C          (R*8)  GAMMA() = INPUT DATA FILE - SELECTED TRANSITION:
C                           MANTISSA OF: ('IGPOW()' => EXPONENT)
C                            GAMMA VALUES             (CASE ' ' & 'P')
C                            RATE COEFFT.(CM3 SEC-1)(CASE 'H','R' & 'I')
C                           DIMENSION => TEMPERATURE 'SCEF()'
C
C          (C*7)  C7      = USED TO PARSE VALUE FOR XJA()
C          (C*7)  CDELIM  = DELIMITERS FOR INPUT OF DATA FROM HEADERS
C          (C*18) C18     = USED TO PARSE VALUE TO CSTRGA()
C          (C*18) C18T    = COPY OF C18 : UNSATISFACTORY METHOD OF
C                           AVOIDING COMPILER REFERENCE ERROR :
C                           DHB 07.04.95
C          (C*80) CLINE   = CURRENT ENERGY LEVEL INDEX PARAMETER LINE
C          (C*75) STRING  = TAIL STRING OF 1ST DATA LINE FOR PARSING
C          (C*44) STRG1   = TAIL STRING OF LEVEL SPEC LINES FOR PARSING
C          (C*128)BUFFER  = GENERAL STRING BUFFER STORAGE
C          (C*3)  CITPOW()= USED TO PARSE VALUES TO ITPOW()
C          (C*5)  CSCEF() = USED TO PARSE VALUES TO SCEF()
C
C          (L*4)  LDATA   = IDENTIFIES  WHETHER  THE END OF AN  INPUT
C                           SECTION IN THE DATA SET HAS BEEN LOCATED.
C                           (.TRUE. => END OF SECTION REACHED)
C          (L*4)  LTCHR   = .TRUE.  => CURRENT 'TCODE()' = 'H' OR 'R'
C                                                          OR 'I'
```

```
C                        = .FALSE. => CURRENT 'TCODE()'.NE.'H' OR 'R'
C                                                               OR 'I'
C          (L*4)  LTCPR   = .TRUE.  => CURRENT 'TCODE()' = 'P' OR 'R'
C                                                              OR 'I'
C                        = .FALSE. => CURRENT 'TCODE()'.NE. 'P' OR 'R'
C                                                                OR 'I'
C          (L*4)  LERROR  = .TRUE.  => UNTIED LEVEL FOUND
C                        = .FALSE. => ALL LEVELS TIED
C          (L*4)  LTIED() = .TRUE.  => SPECIFIED LEVEL TIED
C                        = .FALSE. => SPECIFIED LEVEL IS UNTIED
C                          DIMENSION => LEVEL INDEX
C
C
C NOTE:            LTCHR        LTCPR         TCODE()
C                 ----------------------------------
C                 .TRUE.       .TRUE.    =>     'R','I'
C                 .TRUE.       .FALSE.   =>     'H'
C                 .FALSE.      .TRUE.    =>     'P'
C                 .FALSE.      .FALSE.   =>     ' '
C
C        FOR A-VALUES & GAMMA-VALUES ENTRIES LESS THAN 'DZERO' ARE TAKEN
C        AS BEING EQUAL TO DZERO. THIS AFFECTS THE 'AVAL()' AND 'SCOM()'
C        ARRAYS.
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4UNIT     ADAS      FETCH UNIT NUMBER FOR OUTPUT OF MESSAGES
C          R8FCTN     ADAS      CONVERTS FROM CHARACTER TO REAL VARIABLE
C          I4FCTN     ADAS      CONVERTS FROM CHAR. TO INTEGER  VARIABLE
C          XXWORD     ADAS      PARSES A STRING INTO SEPARATE WORDS
C                               FOR ' ()<>{}' DELIMITERS
C
C AUTHOR:  HP SUMMERS   (REVISION OF BXDATA BY PE BRIDEN)
C          K1/1/57
C          JET EXT. 4941
C
C DATE:    11/06/92
C
C UPDATE:   9/07/93  HPS - USE NEW VERSIONS OF PARSING ROUTINES
C                          B8PARS AND B8PRS1
C UPDATE:  12/07/93  HPS - REVISE TO CONSISTENCY WITH  BXDATA
C                          AT 25/07/93.
C UPDATE:  11/05/95  HPS - ADDED CPRTA TO PARAMETER LIST.ALTERED
C                          'READ()BUFFER' TO BE CONSISTENT WITH IDL-ADAS
C UPDATE:  13/11/95  DHB - INCREASED LENGTH OF CPRTA FROM 4 TO 9 &
C                          STRING FROM 55 TO 75 IN LINE WITH
C                          MODIFICATIONS TO ACCOMODATE J-RESOLVED
C                          PARENT METASTABLES IN THE DATASETS.
C UPDATE:  16/01/96  DHB - INCREASED LENGTH OF CLINE TO 92 & STRG1 TO
C                          56. ALTERED FORMAT NO. 1003 & READING OF
C                          CLINE FORMAT TO ACCOMODATE CHANGES.
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 19-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - PUT UNDER SCCS CONTROL
C
C VERSION: 1.2 DATE: 03-07-97
C MODIFIED: RICHARD MARTIN
C - CHANGED I3 TO I4 IN FORMAT STATEMENT 1001
```

```
C
C VERSION: 1.3 DATE: 20-11-98
C MODIFIED: DAVID H. BROOKS
C - CHANGED MTIED TO 250.
C
C VERSION: 1.4 DATE: 02-03-2003
C MODIFIED: Martin O'Mullane
C - Changed MTIED TO 1500.
C
C UPDATE: 1.5 DATE: 17/05/07
C MODIFIED: Allan Whiteford
C - Updated comments as part of subroutine documentation
C                 procedure.
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      CHARACTER           CPLA(NDLEV)
      CHARACTER*9         CPRTA(NDMET)
      CHARACTER*(*)       CSTRGA(NDLEV)
      CHARACTER           TCODE(NDTRN)
      CHARACTER*3         TITLED
      INTEGER             I1A(NDTRN),  I2A(NDTRN),  IA(NDLEV),    IL
      INTEGER             ILA(NDLEV),  IPLA(NDMET,NDLEV)
      INTEGER             ISA(NDLEV),  ITRAN,       IUNIT,        IZ
      INTEGER             IZ0,         IZ1,         MAXLEV,       NDLEV
      INTEGER             NDMET,       NDTRN,       NPL
      INTEGER             NPLA(NDLEV), NV
      LOGICAL             LBSETA(NDMET)
      REAL*8              AVAL(NDTRN), BWNO,        BWNOA(NDMET)
      REAL*8              PRTWTA(NDMET),            SCEF(NVMAX)
      REAL*8              SCOM(NVMAX,NDTRN),        WA(NDLEV)
      REAL*8              XJA(NDLEV),  ZPLA(NDMET,NDLEV)
```

3.68 bapars: Subroutine bapars from library adas2xx

```
       SUBROUTINE BAPARS(NDMET,STRING,NPT,BWNOA,LSETA,
     &                   PRTWTA,CPRTA,IFAIL)
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: BAPARS *********************
C
C  PURPOSE:  TO ANALYSE THE TAIL CHARACTER STRING OF THE FIRST LINE OF
C            A SPECIFIC ION FILE INTO BINDING WAVE NUMBERS FOR DIFFERENT
C            PARENTS AND STATISTICAL WEIGHTS FOR THE PARENTS.
C            MODIFICATION OF B8PARS.
C
C  CALLING PROGRAM: ADAS210
C
C  NOTES: DETECT  -  BINDING WAVE NUMBER WHICH PRECEED TERM ASSIGNATION
C                 -   TERM CONTAINED IN '(..)'.
C
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  NDMET    =  MAXIMUM NUMBER OF METASTABLES ALLOWED
C  INPUT : (C*(*))STRING   =  STRING TO BE PARSED
C
C  OUTPUT: (I*4)  NPT      =  NUMBER OF BINDING WAVE NUMBERS DETECTED
C  OUTPUT: (L*4)  LSETA()  =  .TRUE.  - PARENT TERM SET FOR THIS W.NO.
C                             .FALSE. - PARENT TERM NOT SET FOR W.NO.
C  OUTPUT: (L*4)  LFND     =  .TRUE.  - L QUANTUM NUMBER PRESENT IN
C                                       STRING
C                             .FALSE. - NO L QUANTUM NUMBER DETECTED
C  OUTPUT: (R*8)  BWNOA()  =  BINDING WAVE NUMBERS
C  OUTPUT: (R*8)  PRTWTA() =  PARENT STATISTICAL WEIGHTS
C  OUTPUT: (C*9)  CPRTA()  =  PARENT NAME IN BRACKETS
C  OUTPUT: (I*4)  IFAIL    =  0 - SUBROUTINE CONCLUDES CORRECTLY
C                             1 - FAULT DETECTED IN SUBROUTINE
C                             2 - SINGLE IONISATION POTENTIAL DETECTED
C
C          (I*4)  MAXWRD   =  MAXIMUM NUMBER OF WORDS SOUGHT INITIALLY
C                             INITIALLY, FINALLY NUMBER ACTUALLY FOUND
C          (I*4)  NFIRST   =  FIRST WORD TO BE EXTRACTED FROM STRING
C          (I*4)  IFIRST() =  INDEX OF FIRST CHAR. OF WORD () IN STRING
C          (I*4)  ILAST()  =  INDEX OF LAST  CHAR. OF WORD () IN STRING
C          (I*4)  IWORDS   =  NUMBER OF WORDS FOUND IN STRING
C          (I*4)  IABT     =  FAILURE NUMBER FROM R8FCTN
C          (I*4)  NCHAR    =  NUMBER OF CHARACTERS IN SUBSTRING
C          (I*4)  I        =  GENERAL USE
C          (I*4)  J        =  GENERAL USE
C          (I*4)  K        =  GENERAL USE
C          (I*4)  IC       =  GENERAL USE
C          (I*4)  ITYPE    =  RESOLUTION OF PARENT METASTABLES
C                             1 - LS RESOLVED
C                             2 - LSJ RESOLVED
C                             3 - ARBITRARY RESOLUTION
C          (I*4)  ITP      =  FLAG FOR INCOMPATIBLE TYPES
C          (I*4)  ITYP     =  COPY OF CURRENT ITYPE
C          (I*4)  KMRK     =  POSITION MARKER IN THE STRING FOR PARENT
C                             L QUANTUM NUMBER
C          (R*8)  TWTA()   =  (2L+1) VALUE FOR PARENT L QUANTUM NUMBER
C          (C*1)  CTRMA()  =  PARENT L QUANTUM NUMBER LETTER SET
```

```
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4UNIT     ADAS      FETCH UNIT NUMBER FOR OUTPUT OF MESSAGES
C          R8FCTN     ADAS      CONVERTS FROM CHARACTER TO REAL VARIABLE
C          I4FCTN     ADAS      CONVERTS FROM CHAR. TO INTEGER  VARIABLE
C          XXWORD     ADAS      PARSES A STRING INTO SEPARATE WORDS
C                               FOR ' ()<>{}' DELIMITERS
C
C AUTHOR:  HP SUMMERS
C          K1/1/57
C          JET EXT. 4941
C
C DATE:    22/06/92
C
C UPDATE:   8/07/93 - HPS  ALTERED TO USE XXWORD PARSING ROUTINE
C
C UPDATE:  11/05/95 - HPS  ADD CPRTA TO PARAMETER LIST
C
C UPDATE:  13/11/95 - DHB  EXTENDED ROUTINE TO HANDLE J/ARBITRARY
C                          RESOLVED PARENT METASTABLE INFORMATION
C                          IN THE FIRST DATA CARD STRING. INTRODUCED
C                          ITYPE TO FLAG RESOLUTION IF REQUIRED.
C
C UPDATE:  21/12/95 - DHB  INCREASED SIZE OF IFIRST & ILAST TO 12 IN
C                          LINE WITH INCREASE TO NDMET
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 22-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - PUT UNDER SCCS CONTROL
C
C VERSION: 1.2                          DATE: 28-1-96
C MODIFIED: HPS + WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C               - PUT IN DEFAULT FOR NO PARENTS IN FILE.
C
C-----------------------------------------------------------------------
      CHARACTER*9         CPRTA(NDMET)
      CHARACTER*(*)       STRING
      INTEGER             IFAIL,       NDMET,       NPT
      LOGICAL             LSETA(NDMET)
      REAL*8              BWNOA(NDMET),             PRTWTA(NDMET)
```

3.69 baprs1: Subroutine baprs1 from library adas2xx

```
       SUBROUTINE BAPRS1(NDMET,STRING,WNO,CPL,NPT,IPLA,ZPLA,IFAIL)
C
C-----------------------------------------------------------------------
C
C  ***************** FORTRAN77 SUBROUTINE: BAPRS1 *********************
C
C  PURPOSE:  TO ANALYSE THE TAIL CHARACTER STRING OF A LEVEL DATA LINE
C            OF A SPECIFIC ION FILE INTO WAVE-NUMBER AND SETS OF
C            (PARENT IDENTIFIER, EFFECTIVE ZETA FOR THE PARENT) PAIRS.
C
C  CALLING PROGRAM: ADAS210
C
C  NOTES: DETECT  -  LEVEL WAVE NUMBER WHICH PRECEEDS FIRST '{'
C                 -  SETS OF   PARENT INDEX CONTAINED IN '{.}'
C                              FOLLOWED BY EFFECTIVE ZETA
C         NB. 'X' AS FIRST PARENT ASSIGNMENT MEANS EXCLUDE IONISATION
C             FROM THIS LEVEL.
C             NO PARENT ASSIGNMENT MEANS TAKE LOWEST PARENT WITH
C             ZETA =1.
C             LOWEST PARENT BUT NO ZETA MEANS TAKE ZETA =1.
C             IF THERE IS MORE THAN ONE PARENT THEN ZETA'S MUST BE IN.
C
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  NDMET    =  MAXIMUM NUMBER OF PARENTS
C  INPUT : (C*(*))STRING   =  STRING TO BE PARSED
C
C  OUTPUT: (R*8)  WNO      =  EXCITATION WAVE NUMBER OF LEVEL RELATIVE
C                             TO LOWEST PARENT
C  OUTPUT: (C*1)  CPL      =  LEAD PARENT FOR IONISATION  OR 'X'
C  OUTPUT: (I*4)  NPT      =  NUMBER OF PARENTS DETECTED
C  OUTPUT: (I*4)  IPLA()   =  PARENT INDICES.
C  OUTPUT: (R*8)  ZPLA()   =  EFFECTIVE ZETA FOR PARENT IPLA()
C  OUTPUT: (I*4)  IFAIL    =  0 - SUBROUTINE CONCLUDES CORRECTLY
C                             1 - FAULT DETECTED IN SUBROUTINE
C                             2 - SINGLE IONISATION POTENTIAL DETECTED
C
C          (I*4)  MAXWRD   =  MAXIMUM NUMBER OF WORDS SOUGHT INITIALLY
C                             INITIALLY, FINALLY NUMBER ACTUALLY FOUND
C          (I*4)  NFIRST   =  FIRST WORD TO BE EXTRACTED FROM STRING
C          (I*4)  IFIRST() =  INDEX OF FIRST CHAR. OF WORD () IN STRING
C          (I*4)  ILAST()  =  INDEX OF LAST  CHAR. OF WORD () IN STRING
C          (I*4)  IWORDS   =  NUMBER OF WORDS FOUND IN STRING
C
C          (L*4)  LSET     =  .TRUE.  -  WAVE NUMBER PART SET
C                             .FALSE. -  WAVE NUMBER PART NOT SET
C          (L*4)  LWNO     =  .TRUE.  -  IN THE WAVE NUMBER PART
C                             .FALSE. -  NOT IN THE WAVE NUMBER PART
C          (L*4)  LPRNT    =  .TRUE.  -  IN A PARENT SPECIFIER
C                             .FALSE. -  NOT IN A PARENT SPECIFIER
C          (L*4)  LZETA    =  .TRUE.  -  IN A ZETA SPECIFIER
C                             .FALSE. -  NOT IN A ZETA SPECIFIER
C          (I*4)  IC       =  GENERAL USE
C          (I*4)  IABT     =  FAILURE NUMBER FROM R8FCTN
C          (I*4)  NCHAR    =  NUMBER OF CHARACTERS IN SUBSTRING
C          (C*15) SSTRNG   =  ISOLATED SUBSTRING
C
C ROUTINES:
```

```
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4UNIT     ADAS      FETCH UNIT NUMBER FOR OUTPUT OF MESSAGES
C          R8FCTN     ADAS      CONVERTS FROM CHARACTER TO REAL VARIABLE
C          I4FCTN     ADAS      CONVERTS FROM CHAR. TO INTEGER  VARIABLE
C          XXWORD     ADAS      PARSES A STRING INTO SEPARATE WORDS
C                               FOR ' ()<>{}' DELIMITERS
C
C AUTHOR:  HP SUMMERS
C          K1/1/57
C          JET EXT. 4941
C
C DATE:    22/06/92
C
C UPDATE:   8/07/93 - HPS  ALTERED TO USE XXWORD PARSING ROUTINE
C
C UPDATE:  21/12/95 - DHB  INCREASED SIZE OF IFIRST & ILAST TO 12 IN
C                          LINE WITH CHANGE TO NDMET
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 22-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - PUT UNDER SCCS CONTROL
C
C-----------------------------------------------------------------------
      CHARACTER           CPL
      CHARACTER*(*)       STRING
      INTEGER             IFAIL,       IPLA(NDMET), NDMET,       NPT
      REAL*8              WNO,         ZPLA(NDMET)
```

3.70 baubnd: Subroutine baubnd from library adas2xx

```
       SUBROUTINE BAUBND( ITRAN  , ITRANB , ITRAN2 , I1A   , I2A   ,
     &                    I1BA   , I2BA   , I1A2   , I2A2  , AVAL  ,
     &                    AVALB  , SCOM   , SCOMB  , SCOM2 , NV2   ,
     &                    NDTRN  , NVMAX  , TCODE  , TCODEB, TCODE2,
     &                    INDBL  , NJLEVX , SCOMU  , TCODEU, I1UA  ,
     &                    I2UA   , PRERAT , ILA2   , ISA2  , XJA2  ,
     &                    NDLEV  , IL2    , BNDLS  , NCHK  , IUA   ,
     &                    ILUA   , ISUA   , CSTRGUA, WUA   , XJUA  ,
     &                    IA     , ILA    , ISA    , CSTRGS, WA    ,
     &                    XJA    , IA2    , CSTRGA2, WA2   , NBLEVX,
     &                    INDUL  , NCHKU  , ISORT  , INDBS , AVALU ,
     &                    AVAL2  , PREA   , ITRANU , IUL   , XLSA  ,
     &                    BNDPR  , NBCPRT , IL3    , CPRTAU, IA3   ,
     &                    ILA3   , ISA3   , XJA3   , WA3   , BWNO2 ,
     &                    NPL2   , BWNOA2 , PRTWTA2, CPRTA2, NDMET ,
     &                    IPMDFLG, BWNOAU , CPLA2  , NPLA2 , IPLA2 ,
     &                    ZPLA2  , CPLAU  , NPLAU  , IPLAU , ZPLAU ,
     &                    IMRK   , PRTWTAU, IRCHK  , NZEROS)
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: BAUBND *********************
C
C  PURPOSE: TO UNBUNDLE A SPECIFIC ION FILE ACCORDING TO THE SPLIT UP
C           FRACTIONS OBTAINED FROM A SUPERSTRUCTURE FILE, FILLING IN
C           WITH THE STATISTICAL METHOD OF SARAPH, SEATON & SHEMMING
C           (1969), WHEN NO DATA IS AVAILABLE.
C
C  CALLING PROGRAM: ADAS210
C
C  SUBROUTINE:
C
C  INPUT:
C            (I*4)  NDLEV   = MAXIMUM NUMBER OF LEVELS THAT CAN BE READ
C            (I*4)  NDTRN   = MAX. NUMBER OF TRANSITIONS THAT CAN BE READ
C            (I*4)  NDMET   = MAX. NUMBER OF METASTABLES ALLOWED
C            (I*4)  NVMAX   = MAX. NUMBER OF TEMPERATURES
C            (I*4)  ITRAN   = SUPERSTRUCTURE FILE: NO. OF TRANSITIONS
C            (I*4)  ITRANB  = BUNDLED SUPERSTRUCTURE FILE: NO. OF TRANSITIONS
C            (I*4)  ITRAN2  = INPUT DATA FILE: NO. OF TRANSITIONS
C            (I*4)  I1A()   = TRANSITION: IN SUPERSTRUCTURE FILE
C                             LOWER ENERGY LEVEL INDEX (CASE ' ' & 'P')
C                             SIGNED PARENT INDEX (CASE 'H','R' & 'I')
C            (I*4)  I2A()   = TRANSITION: IN SUPERSTRUCTURE FILE
C                             UPPER ENERGY LEVEL INDEX (CASE ' ' & 'P')
C                             CAPTURING LEVEL INDEX (CASE 'H','R' & 'I')
C            (I*4)  I1BA()  = TRANSITION: IN BUNDLED SUPERSTRUCTURE FILE
C                             LOWER ENERGY LEVEL INDEX (CASE ' ' & 'P')
C                             SIGNED PARENT INDEX (CASE 'H','R' & 'I')
C            (I*4)  I2BA()  = TRANSITION: IN BUNDLED SUPERSTRUCTURE FILE
C                             UPPER ENERGY LEVEL INDEX (CASE ' ' & 'P')
C                             CAPTURING LEVEL INDEX (CASE 'H','R' & 'I')
C            (I*4)  I1A2()  = TRANSITION: IN INPUT DATA FILE
C                             LOWER ENERGY LEVEL INDEX (CASE ' ' & 'P')
C                             SIGNED PARENT INDEX (CASE 'H','R' & 'I')
C            (I*4)  I2A2()  = TRANSITION: IN INPUT DATA FILE
C                             UPPER ENERGY LEVEL INDEX (CASE ' ' & 'P')
C                             CAPTURING LEVEL INDEX (CASE 'H','R' & 'I')
C            (R*8)  AVAL()  = TRANSITION: IN SUPERSTRUCTURE FILE
```

```
C                          A-VALUE (SEC-1)          (CASE ' ')
C                          NEUTRAL BEAM ENERGY      (CASE 'H')
C                          NOT USED             (CASE 'P','R' & 'I')
C           (R*8)  AVALB() = TRANSITION: IN BUNDLED SUPERSTRUCTURE FILE
C                          A-VALUE (SEC-1)          (CASE ' ')
C                          NEUTRAL BEAM ENERGY      (CASE 'H')
C                          NOT USED             (CASE 'P','R' & 'I')
C           (R*8)  AVAL2() = TRANSITION: IN INPUT DATA FILE
C                          A-VALUE (SEC-1)          (CASE ' ')
C                          NEUTRAL BEAM ENERGY      (CASE 'H')
C                          NOT USED             (CASE 'P','R' & 'I')
C           (R*8)  SCOM(,) = TRANSITION: IN SUPERSTRUCTURE FILE
C                          GAMMA VALUES             (CASE ' ' & 'P')
C                          RATE COEFFT.(CM3 SEC-1)(CASE 'H','R' & 'I')
C                         1ST DIMENSION - TEMPERATURE 'SCEF()'
C                         2ND DIMENSION - TRANSITION NUMBER
C           (R*8)  SCOMB(,)= TRANSITION: IN BUNDLED SUPERSTRUCTURE FILE
C                          GAMMA VALUES             (CASE ' ' & 'P')
C                          RATE COEFFT.(CM3 SEC-1)(CASE 'H','R' & 'I')
C                         1ST DIMENSION - TEMPERATURE 'SCEF()'
C                         2ND DIMENSION - TRANSITION NUMBER
C           (R*8)  SCOM2(,)= TRANSITION: IN INPUT DATA FILE
C                          GAMMA VALUES             (CASE ' ' & 'P')
C                          RATE COEFFT.(CM3 SEC-1)(CASE 'H','R' & 'I')
C                         1ST DIMENSION - TEMPERATURE 'SCEF()'
C                         2ND DIMENSION - TRANSITION NUMBER
C           (C*1)  TCODE() = TRANSITION: DATA TYPE POINTER:
C                         ' ' => Electron Impact   Transition
C                         'P' => Proton   Impact   Transition
C                         'H' => Charge   Exchange Recombination
C                         'R' => Free     Electron Recombination
C                         'I' => Coll. ionisation from lower stage ion
C                          IN SUPERSTRUCTURE FILE
C           (C*1)  TCODEB()= TRANSITION: DATA TYPE POINTER:
C                          IN BUNDLED SUPERSTRUCTURE FILE - SAME CODES
C                          AS TCODE ABOVE
C           (C*1)  TCODE2()= TRANSITION: DATA TYPE POINTER:
C                          IN INPUT DATA FILE - SAME CODES
C                          AS TCODE ABOVE
C           (I*4)  NV2     = INPUT DATA FILE: NUMBER OF GAMMA/TEMPERATURE
C                          PAIRS FOR A GIVEN TRANSITION.
C           (I*4)  INDBL() = VECTOR CONTAINING THE BUNDLED SUPERSTRUCTURE
C                          FILE INDICES AT THE ORIGINAL INDEX LOCATIONS
C           (I*4)  NJLEVX  = THE NO. OF LEVELS IN THE SUPERSTRUCTURE FILE
C           (I*4)  IA()    = SUPERSTRUCTURE FILE ENERGY LEVEL INDEX NUMBER
C           (C*18) CSTRGS()= NOMENCLATURE/CONFIGURATION FOR LEVEL 'IA()'
C                          CONVERTED TO STANDARD FROM EISNER FORM
C           (I*4)  ISA()   = MULTIPLICITY FOR LEVEL 'IA()'
C                          NOTE: (ISA-1)/2 = QUANTUM NUMBER (S)
C           (I*4)  ILA()   = QUANTUM NUMBER (L) FOR LEVEL 'IA()'
C           (R*8)  XJA()   = QUANTUM NUMBER (J-VALUE) FOR LEVEL 'IA()'
C                          NOTE: (2*XJA)+1 = STATISTICAL WEIGHT
C           (R*8)  WA()    = ENERGY RELATIVE TO LEVEL 1 (CM-1) FOR LEVEL
C                         'IA()'
C           (I*4)  IL2     = INPUT DATA FILE: NUMBER OF ENERGY LEVELS
C           (I*4)  IA2()   = INPUT DATA ENERGY LEVEL INDEX NUMBER
C           (C*18) CSTRGA2()= NOMENCLATURE/CONFIGURATION FOR LEVEL 'IA2()'
C           (I*4)  ISA2()  = MULTIPLICITY FOR LEVEL 'IA2()'
C                          NOTE: (ISA2-1)/2 = QUANTUM NUMBER (S)
C           (I*4)  ILA2()  = QUANTUM NUMBER (L) FOR LEVEL 'IA2()'
C           (R*8)  XJA2()  = QUANTUM NUMBER (J-VALUE) FOR LEVEL 'IA2()'
```

```
C                              NOTE: (2*XJA2)+1 = STATISTICAL WEIGHT
C            (R*8)  WA2()    = ENERGY RELATIVE TO LEVEL 1 (CM-1) FOR LEVEL
C                             'IA2()'
C            (I*4)  IL3      = INPUT DATA FILE: NUMBER OF ENERGY LEVELS
C            (I*4)  IA3()    = PARENT SUPERSTRUCTURE ENERGY LEVEL INDEX
C                              NUMBER
C            (C*18) CPRTAU()= NOMENCLATURE/CONFIGURATION FOR NEW PARENTS
C            (I*4)  ISA3()   = MULTIPLICITY FOR LEVEL 'IA3()'
C                              NOTE: (ISA3-1)/2 = QUANTUM NUMBER (S)
C            (I*4)  ILA3()   = QUANTUM NUMBER (L) FOR LEVEL 'IA3()'
C            (R*8)  XJA3()   = QUANTUM NUMBER (J-VALUE) FOR LEVEL 'IA3()'
C                              NOTE: (2*XJA3)+1 = STATISTICAL WEIGHT
C            (R*8)  WA3()    = ENERGY RELATIVE TO LEVEL 1 (CM-1) FOR LEVEL
C                             'IA3()'
C            (I*4)  BNDLS()  = LEVEL/TERM SELECTION VECTOR
C            (I*4)  BNDPR()  = PARENT METASTABLE SELECTION VECTOR
C            (I*4)  NBLEVX   = THE NO. OF LEVELS IN THE BUNDLED
C                              SUPERSTRUCTURE FILE
C            (R*8)  XLSA()   = QUANTUM NUMBER (J-VALUE) FOR BUNDLED
C                              SUPERSTRUCTURE LEVEL 'I2BA()'
C                              NOTE: (2*XLSA)+1 = STATISTICAL WEIGHT
C            (I*4)  NBCPRT   = NUMBER OF SELECTED CONTRIBUTIONS TO PARENTS
C            (R*8)  BWNO2    = IONISATION POTENTIAL (CM-1) OF LOWEST PARENT
C                              IN INPUT DATA FILE
C            (I*4)  NPL2     = NUMBER OF PARENTS ON FIRST LINE OF INPUT
C                              DATA FILE AND USED IN LEVEL ASSIGNMENTS
C            (R*8)  BWNOA2()= IONISATION POTENTIAL (CM-1) OF PARENTS
C                              IN INPUT DATA FILE
C            (R*8)  PRTWTA2()= PARENT WEIGHT FOR BWNOA2()
C            (C*9)  CPRTA2()= PARENT NAME IN BRACKETS IN INPUT DATA FILE
C            (C*1)  CPLA2()  = CHAR. SPECIFYING 1ST PARENT FOR LEVEL 'IA2()'
C                                 INTEGER - PARENT IN BWNOA2() LIST
C                                 'BLANK' - PARENT BWNOA2(1)
C                                   'X'   - DO NOT ASSIGN A PARENT
C            (I*4)  NPLA2()  = NO. OF PARENT/ZETA CONTRIBUTIONS TO IONIS.
C                              OF LEVEL IN INPUT DATA FILE
C            (I*4)  IPLA2(,)= PARENT INDEX FOR CONTRIBUTIONS TO IONIS.
C                              OF LEVEL IN INPUT DATA FILE
C                             1ST DIMENSION: PARENT INDEX
C                             2ND DIMENSION: LEVEL INDEX
C            (I*4)  ZPLA2(,)= EFF. ZETA PARAM. FOR CONTRIBUTIONS TO IONIS.
C                              OF LEVEL IN INPUT DATA FILE
C                             1ST DIMENSION: PARENT INDEX
C                             2ND DIMENSION: LEVEL INDEX
C            (I*4)  IPMDFLG = FLAG FOR PARENT SUPERSTRUCTURE FILE
C                              AVAILABILITY
C
C  OUTPUT:
C            (I*4)  ITRANU   = OUTPUT DATA FILE: NO. OF TRANSITIONS
C            (I*4)  I1UA()   = TRANSITION: IN OUTPUT DATA FILE
C                              LOWER ENERGY LEVEL INDEX (CASE ' ' & 'P')
C                              SIGNED PARENT INDEX (CASE 'H','R' & 'I')
C            (I*4)  I2UA()   = TRANSITION: IN OUTPUT DATA FILE
C                              UPPER ENERGY LEVEL INDEX (CASE ' ' & 'P')
C                              CAPTURING LEVEL INDEX (CASE 'H','R' & 'I')
C            (R*8)  AVALU() = TRANSITION: IN OUTPUT DATA FILE
C                              A-VALUE (SEC-1)          (CASE ' ')
C                              NEUTRAL BEAM ENERGY      (CASE 'H')
C                              NOT USED             (CASE 'P','R' & 'I')
C            (R*8)  SCOMU(,)= TRANSITION: IN OUTPUT DATA FILE
C                              GAMMA VALUES             (CASE ' ' & 'P')
```

```
C                              RATE COEFFT.(CM3 SEC-1)(CASE 'H','R' & 'I')
C                             1ST DIMENSION - TEMPERATURE 'SCEF()'
C                             2ND DIMENSION - TRANSITION NUMBER
C            (C*1)  TCODEU()= TRANSITION: DATA TYPE POINTER:
C                              IN OUTPUT DATA FILE - SAME CODES
C                              AS TCODE ABOVE
C            (R*8)  PRERAT(,)=ARRAY OF PREMULTIPLIERS FOR THE J-RESOLVED
C                              LEVELS. RATIO OF INPUT DATA TO BUNDLED DATA
C                              FOR A TRANSITION
C                             1ST DIMENSION - TEMPERATURE 'SCEF()'
C                             2ND DIMENSION - TRANSITION NUMBER
C            (R*8)  PREA()  = PREMULTIPLIERS FOR THE SUPERSTRUCTURE
C                              A-VALUES. RATIO OF STAT. WEIGHTED INPUT
C                              DATA TO BUNDLED DATA.
C            (I*4)  IUA()   = ENERGY LEVEL INDEX NUMBER
C            (C*18) CSTRGUA()= NOMENCLATURE/CONFIGURATION FOR LEVEL 'IUA()'
C            (I*4)  ISUA()  = MULTIPLICITY FOR LEVEL 'IUA()'
C                              NOTE: (ISUA-1)/2 = QUANTUM NUMBER (S)
C            (I*4)  ILUA()  = QUANTUM NUMBER (L) FOR LEVEL 'IUA()'
C            (R*8)  XJUA()  = QUANTUM NUMBER (J-VALUE) FOR LEVEL 'IUA()'
C                              NOTE: (2*XJUA)+1 = STATISTICAL WEIGHT
C            (R*8)  WUA()   = ENERGY RELATIVE TO LEVEL 1 (CM-1) FOR LEVEL
C                             'IUA()'
C            (I*4)  INDUL() = VECTOR CONTAINING THE UNBUNDLED FILE
C                              INDICES AT THE ORIGINAL INDEX LOCATIONS
C            (I*4)  IUL     = OUTPUT DATA FILE: NUMBER OF ENERGY LEVELS
C            (R*8)  BWNOAU()= IONISATION POTENTIAL (CM-1) OF PARENTS
C                              IN OUTPUT DATA FILE
C            (R*8)  PRTWTAU()= PARENT WEIGHT FOR BWNOAU()
C            (C*1)  CPLAU() = CHAR. SPECIFYING 1ST PARENT FOR LEVEL 'IUA()'
C                                INTEGER - PARENT IN BWNOA2() LIST
C                                'BLANK' - PARENT BWNOA2(1)
C                                  'X'   - DO NOT ASSIGN A PARENT
C            (I*4)  NPLAU() = NO. OF PARENT/ZETA CONTRIBUTIONS TO IONIS.
C                              OF LEVEL IN OUTPUT DATA FILE
C            (I*4)  IPLAU(,)= PARENT INDEX FOR CONTRIBUTIONS TO IONIS.
C                              OF LEVEL IN OUTPUT DATA FILE
C                             1ST DIMENSION: PARENT INDEX
C                             2ND DIMENSION: LEVEL INDEX
C            (I*4)  ZPLAU(,)= EFF. ZETA PARAM. FOR CONTRIBUTIONS TO IONIS.
C                              OF LEVEL IN OUTPUT DATA FILE
C                             1ST DIMENSION: PARENT INDEX
C                             2ND DIMENSION: LEVEL INDEX
C            (I*4)  NCHK()  = VECTOR NOTING REPEATED USER SELECTIONS
C                              AND HOW OFTEN THEY OCCUR, FOR LEVELS
C            (I*4)  NCHKU() = VECTOR NOTING THE LEVELS SELECTED FOR
C                              UNBUNDLING AND THEIR NEW POSITIONING
C            (I*4)  ISORT() = CROSS REFERENCE VECTOR FOR NEW INDEXING
C            (I*4)  INDBS() = CROSS REFERENCE VECTOR FOR NEW BUNDLED
C                              SUPERSTRUCTURE INDEXING
C            (I*4)  IMRK()  = VECTOR NOTING REPEATED USER SELECTIONS
C                              AND HOW OFTEN THEY OCCUR, FOR PARENTS
C            (I*4)  IRCHK() = VECTOR NOTING THE PARENTS SELECTED FOR
C                              UNBUNDLING AND THEIR NEW POSITIONING
C
C  ROUTINES: NONE
C
C  AUTHOR:   DAVID H.BROOKS (UNIV.OF STRATHCLYDE) EXT.4213/4205
C
C  DATE:     12/01/96
C
```

```
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 22-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - PUT UNDER SCCS CONTROL
C
C VERSION: 1.2                          DATE: 26-1-96
C MODIFIED: DAVID H.BROOKS
C               - ALTERED MATCHING OF LSJ LEVELS TO DATASETS IN ORDER
C                 TO ALLOW PARTIAL SPLITTING OF THE SOURCE FILE.
C
C VERSION: 1.3                          DATE: 19-11-98
C MODIFIED: DAVID H.BROOKS
C               - MODIFIED TO ALLOW EXTRA LEVELS TO BE INTERSPERSED
C                 WITH THE ONES BEING USED. THESE ARE OMITTED FROM
C                 THE ACTUAL CALCULATIONS BY NOTING THEIR POSITIONS
C                 IN BNDLS.
C
C-----------------------------------------------------------------------
      CHARACTER           CPLA2(NDLEV),             CPLAU(NDLEV)
      CHARACTER*9         CPRTA2(NDMET),            CPRTAU(2*NDMET)
      CHARACTER*18        CSTRGA2(NDLEV),           CSTRGS(NDLEV)
      CHARACTER*18        CSTRGUA(NDLEV)
      CHARACTER           TCODE(NDTRN),             TCODE2(NDTRN)
      CHARACTER           TCODEB(NDTRN),            TCODEU(NDTRN)
      INTEGER             BNDLS(NDLEV),             BNDPR(NDLEV)
      INTEGER             I1A(NDTRN),  I1A2(NDTRN), I1BA(NDTRN)
      INTEGER             I1UA(NDTRN), I2A(NDTRN),  I2A2(NDTRN)
      INTEGER             I2BA(NDTRN), I2UA(NDTRN), IA(NDLEV)
      INTEGER             IA2(NDLEV),  IA3(NDLEV),  IL2,          IL3
      INTEGER             ILA(NDLEV),  ILA2(NDLEV), ILA3(NDLEV)
      INTEGER             ILUA(NDLEV), IMRK(2*NDMET)
      INTEGER             INDBL(NJLEVX),            INDBS(NDLEV)
      INTEGER             INDUL(NDLEV),             IPLA2(NDMET,NDLEV)
      INTEGER             IPLAU(2*NDMET,NDLEV),     IPMDFLG
      INTEGER             IRCHK(2*NDMET),           ISA(NDLEV)
      INTEGER             ISA2(NDLEV), ISA3(NDLEV), ISORT(NDLEV)
      INTEGER             ISUA(NDLEV), ITRAN,       ITRAN2,       ITRANB
      INTEGER             ITRANU,      IUA(NDLEV),  IUL,          NBCPRT
      INTEGER             NBLEVX,      NCHK(NJLEVX)
      INTEGER             NCHKU(NDLEV),             NDLEV,        NDMET
      INTEGER             NDTRN,       NJLEVX,      NPL2
      INTEGER             NPLA2(NDLEV),             NPLAU(NDLEV)
      INTEGER             NV2,         NVMAX,       NZEROS
      REAL*8              AVAL(NDTRN), AVAL2(NDTRN)
      REAL*8              AVALB(NDTRN),             AVALU(NDTRN)
      REAL*8              BWNO2,       BWNOA2(NDMET)
      REAL*8              BWNOAU(2*NDMET),          PREA(NDTRN)
      REAL*8              PRERAT(NVMAX,NDTRN),      PRTWTA2(NDMET)
      REAL*8              PRTWTAU(2*NDMET),         SCOM(NVMAX,NDTRN)
      REAL*8              SCOM2(NVMAX,NDTRN),       SCOMB(NVMAX,NDTRN)
      REAL*8              SCOMU(NVMAX,NDTRN),       WA(NDLEV)
      REAL*8              WA2(NDLEV),  WA3(NDLEV),  WUA(NDLEV)
      REAL*8              XJA(NDLEV),  XJA2(NDLEV), XJA3(NDLEV)
      REAL*8              XJUA(NDLEV), XLSA(NDLEV), ZPLA2(NDMET,NDLEV)
      REAL*8              ZPLAU(2*NDMET,NDLEV)
```

3.71 bbitrp: Subroutine bbitrp from library adas2xx

```
      subroutine bbitrp( ninmx  , noutmx ,
     &                   nein   , neout  , te     ,
     &           nform1 , param1 , nform2 , param2 ,
     &  ein    , fin    , eout   , fout
     &         )

C-----------------------------------------------------------------------
C
C    ****************** fortran77 subroutine: bbitrp ******************
C
C  purpose: To interpolate/extrapolate numerical distribution from
C           fin(ein) to fout(eout).
C           A f=sqrt(E)*exp(-E) fit is chosen for interpolation
C           Extrapolation uses limit behaviour from nform1 and nform2
C
C  calling program: adas211
C
C  input : (i*4)  ninmx    = max no of input energies
C  input : (i*4)  noutmx   = max no of output energies
C  input : (i*4)  nein     = no of input energies
C  input : (i*4)  neout    = no of output energies
C  input : (r*8)  te       = temperature
C  input : (i*4)  nform1   = type of threshold behaviour
C                              1 => cutoff
C                              2 => energy^param1
C  input : (r*8)  param1   = parameter of threshold form
C  input : (i*4)  nform2   = type of high-energy behaviour
C                              1 => cutoff
C                              2 => energy^-param2(1)
C                              3 => exp(-param2(1)*energy)
C                              4 => exp(-param2(1)*energy^param2(2))
C  input : (r*8)  param2() = parameter of high-energy form
C
C  input : (r*8)  ein()    = input energy of distribution
C  input : (r*8)  fin()    = value of distribution at ein
C  input:  (r*8)  eout()   = output energy
C
C  output: (r*8)  fout()   = (value of distribution at eout)/sqrt(eout)
C
C
C  author: Paul Bryans, University of Strathclyde
C
C  date:   30/11/04
C
C  update:
C
C-----------------------------------------------------------------------
      INTEGER             NEIN,         NEOUT,       NFORM1,      NFORM2
      INTEGER             NINMX,        NOUTMX
      REAL*8              EIN(NINMX),   EOUT(NOUTMX)
      REAL*8              FIN(NINMX),   FOUT(NOUTMX),             PARAM1
      REAL*8              PARAM2(2),    TE
```

3.72 bbprs1: Subroutine bbprs1 from library adas2xx

```
       SUBROUTINE BBPRS1(NDMET,STRING,WNO,CPL,NPT,IPLA,ZPLA)
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: BBPRS1 *********************
C
C  PURPOSE:  TO ANALYSE THE TAIL CHARACTER STRING OF A LEVEL DATA LINE
C            OF A SPECIFIC ION FILE INTO WAVE-NUMBER AND SETS OF
C            (PARENT IDENTIFIER, EFFECTIVE ZETA FOR THE PARENT) PAIRS.
C
C  CALLING PROGRAM: ADAS208
C
C  NOTES: DETECT  -  LEVEL WAVE NUMBER WHICH PRECEEDS FIRST '{'
C                 -  SETS OF   PARENT INDEX CONTAINED IN '{.}'
C                              FOLLOWED BY EFFECTIVE ZETA
C         NB. 'X' AS FIRST PARENT ASSIGNMENT MEANS EXCLUDE IONISATION
C             FROM THIS LEVEL.
C             NO PARENT ASSIGNMENT MEANS TAKE LOWEST PARENT WITH
C             ZETA =1.
C             LOWEST PARENT BUT NO ZETA MEANS TAKE ZETA =1.
C             IF THERE IS MORE THAN ONE PARENT THEN ZETA'S MUST BE IN.
C
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  NDMET    =  MAXIMUM NUMBER OF PARENTS
C  INPUT : (C*(*))STRING   =  STRING TO BE PARSED
C
C  OUTPUT: (R*8)  WNO      =  EXCITATION WAVE NUMBER OF LEVEL RELATIVE
C                             TO LOWEST PARENT
C  OUTPUT: (C*1)  CPL      =  LEAD PARENT FOR IONISATION  OR 'X'
C  OUTPUT: (I*4)  NPT      =  NUMBER OF BINDING WAVE NUMBERS DETECTED
C  OUTPUT: (I*4)  IPLA()   =  PARENT INDICES.
C  OUTPUT: (R*8)  ZPLA()   =  EFFECTIVE ZETA FOR PARENT IPLA()
C
C          (L*4)  LSET     =  .TRUE.  -  WAVE NUMBER PART SET
C                             .FALSE. -  WAVE NUMBER PART NOT SET
C          (L*4)  LWNO     =  .TRUE.  -  IN THE WAVE NUMBER PART
C                             .FALSE. -  NOT IN THE WAVE NUMBER PART
C          (L*4)  LPRNT    =  .TRUE.  -  IN A PARENT SPECIFIER
C                             .FALSE. -  NOT IN A PARENT SPECIFIER
C          (L*4)  LZETA    =  .TRUE.  -  IN A ZETA SPECIFIER
C                             .FALSE. -  NOT IN A ZETA SPECIFIER
C          (I*4)  IC       =  GENERAL USE
C          (I*4)  IABT     =  FAILURE NUMBER FROM R8FCTN
C          (I*4)  NCHAR    =  NUMBER OF CHARACTERS IN SUBSTRING
C          (C*15) SSTRNG   =  ISOLATED SUBSTRING
C
C ROUTINES: NONE
C
C AUTHOR:  HP SUMMERS
C          K1/1/57
C          JET EXT. 4941
C
C DATE:    22/06/92
C
C UPDATE:  24/06/96  HP SUMMERS - CHANGED SOUBRTOUNE NAME FROM B8PRS1 TO
C                                 BBPRS1
C
C UNIX-IDL PORT:
```

```
C
C AUTHOR:  WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C
C DATE:    4TH JULY 1996
C
C VERSION: 1.1                          DATE: 04-07-96
C MODIFIED: WILLIAM OSBORN
C               - FIRST VERSION.
C
C VERSION: 1.2     DATE: 17-03-03
C MODIFIED: RICHARD MARTIN
C INITIALISED LWNO AS .FALSE.
C
C-----------------------------------------------------------------------
      CHARACTER           CPL
      CHARACTER*(*)       STRING
      INTEGER             IPLA(NDMET), NDMET,       NPT
      REAL*8              WNO,         ZPLA(NDMET)
```

3.73 bbprs3: Subroutine bbprs3 from library adas2xx

```
       SUBROUTINE BBPRS3( STRING, IA, LCLSHL )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: BBPRS3 *********************
C
C  PURPOSE:  TO ANALYSE A CONFIGURATION CHARACTER STRING INTO A  INTEGER
C            ARRAY OF OCCUPATION NUMBERS IN A STANDARD ORDER
C
C  CALLING PROGRAM: ADAS211
C
C  NOTES: THE STANDARD ORDER IS 1S,2S,2P,3S,3P,3D ......, 4F (15 VALUES)
C         CLOSED SHELLS WITHIN THE ACTIVE N-SHELLS ARE ASSUMED FULLY
C         OCCUPIED
C
C
C  SUBROUTINE:
C
C  INPUT : (C*(*))STRING   =  STRING TO BE PARSED
C          (L*4)  LCLSHL   =  SWITCH ON CLOSED SHELL ASSUMPTION
C
C  OUTPUT: (I*4)  IA()     =  SET OF OCCUPATION NUMBERS IN STANDARD
C                             ORDER
C
C ROUTINES: NONE
C
C AUTHOR:  HP SUMMERS
C          K1/1/57
C          JET EXT. 4941
C
C DATE:    29/06/92
C
C UPDATE:  W.J. DICKSON  7/10/92
C          ADDED PARAMETER LCLSHL TO SWITCH OFF CLOSED SHELL
C          APPROXIMATION
C UPDATE:  H. P. SUMMERS  1/10/96
C          PERMITTED LOWER AND UPPER CASE ORBITAL L-VALUES
C          IN CONFIGURATION STRINGS. DETECT RETURNED L<0
C
C UNIX-IDL PORT:
C
C AUTHOR:  WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C
C DATE:    4TH JULY 1996
C
C VERSION: 1.1                          DATE: 04-07-96
C MODIFIED: WILLIAM OSBORN
C               - FIRST VERSION.
C
C VERSION: 1.2                          DATE: 14-10-96
C MODIFIED: WILLIAM OSBORN
C               - ADDED CHANGES DATED 1/10/96 ABOVE
C
C VERSION: 1.3                          DATE: 17-05-07
C MODIFIED: Allan Whiteford
C               - Removed non-standard control character from
C                 comments.
C
C-----------------------------------------------------------------------
      CHARACTER*(*)       STRING
```

```
INTEGER             IA(15)
LOGICAL             LCLSHL
```

3.74 bbrint: Subroutine bbrint from library adas2xx

```
      subroutine bbrint( ndgnt  ,
     &     gaunt  , vve    , z1     , v    ,
     &          maxe   , temp   , dparam , dist ,
     &  f      , rrcint
     & )

C-----------------------------------------------------------------------
C
C    ****************** fortran77 subroutine: bbrint ******************
C
C  purpose: To calculate radiative recombination coefficient when
C           electron distribution is not Maxwellian.
C
C  calling program: adas211
C
C  input : (i*4)  ndgnt   = max no of vve gaunt pairs
C  input : (r*8)  gaunt() = the bound-free gaunt factor
C  input : (r*8)  vve()   = v**2*e
C                           where e=(free electron energy)/z1**2 (ryd)
C  input : (r*8)  z1      = parent ion charge
C  input : (r*8)  v       = effective principal quantum number
C  input : (i*4)  maxe    = number of vve gaunt pairs
C  input : (r*8)  temp    = effective temperature (kelvin),
C                           2/3 * mean energy of distribution
C  input : (r*8)  dparam  = parameter describing distribution function:
C                             kappa dist.       => kappa
C                             Druyvesteyn dist. => x
C  input : (i*4)  dist    = non-Maxwellian distribution type:
C                             1 => kappa distribution
C                             2 => numerical distribution
C                             3 => Druyvesteyn distribution
C  input : (r*8)  f()     = numerical distribution function at vve
C
C  output: (r*8)  rrcint  = radiative recombination coefficient (cm3 sec-1)
C
C  local : (r*8)  ryd     = Rydberg constant (eV)
C  local : (r*8)  te      = effective temperature (eV)
C  local : (r*8)  ip      = ionisation potential (eV)
C  local : (r*8)  kek     = kappa * characteristic energy of kappa dist.
C  local : (r*8)  ex      = characteristic energy of Druyvesteyn dist.
C  local : (r*8)  alpha   = fine structure constant
C  local : (r*8)  c       = speed of light in vacuum (cm sec-1)
C  local : (r*8)  a0      = Bohr radius (cm)
C  local : (r*8)  energy()= free electron energy (eV)
C  local : (r*8)  int1    = integrand at energy(i)
C  local : (r*8)  int2    = integrand at energy(i+1)
C  local : (r*8)  de      = energy difference from i to i+1
C
C  routines:
C          routine    source    brief description
C          ------------------------------------------------------------
C          lngama               evaluates ln(gamma(x))
C
C  author: Paul Bryans, University of Strathclyde
C
C  date:   23/01/04
C
C  update: 26/01/04 - Paul Bryans
C          added Druyvesteyn distribution (dist = 3)
```

```
C
C  update: 02/12/04 - Paul Bryans
C          added numerical distribution (dist = 2)
C
C  update: 02/02/05 - Allan Whiteford
C          Declared i4unit as an integer.
C
C  update: 20/07/07 - Allan Whiteford
C          Removed commet stating that Druyvesteyn and numerical
C          distributions can't be handled.
C
C-----------------------------------------------------------------------
      INTEGER             DIST,         MAXE,         NDGNT
      REAL*8              DPARAM,       F(NDGNT),     GAUNT(NDGNT)
      REAL*8              RRCINT,       TEMP,         V
      REAL*8              VVE(NDGNT),   Z1
```

3.75 bbspln: Subroutine bbspln from library adas2xx

```
      subroutine bbspln( ndtem   , ntmax   ,
     &                   nblock  , maxt    ,
     &                   tin     , tout    ,
     &                   rrcin   , rrcout
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** fortran77 subroutine: bbspln *********************
C
C  purpose:
C    1) performs cubic spline on log(temp) versus log(rad.rec.coeff.)
C       input data. ('tin' versus 'rrcin' , nblock data pairs)
C
C    2) interpolates 'maxt'  rrcout values using above splines at
C       temperatures read in from adf08 file for tabular output.

C
C  calling program: adas211
C
C
C  subroutine:
C
C
C  input : (i*4)  ndtem   = maximum number of adf08 temperatures
C  input : (i*4)  ntmax   = maximum number of adf37 temperatures
C  input : (i*4)  nblock  = input data file: number of rrc/temperature
C      pairs read for the transition being assessed
C  input : (i*4)  maxt    = number of adf08 temperature values at
C      which interpolated rrc values are required
C      for tabular output.
C  input : (r*8)  tin()   = adf37 temperatures (kelvin)
C  input : (r*8)  tout()  = adf08 entered temperatures (kelvin)
C  input : (r*8)  rrcin() = rrc values at 'tin()'.
C
C  output: (r*8)  rrcout()= spline interpolated rrc values at 'tout()'
C
C  local : (i*4)  nin   = parameter = max. no. of input temp/rrc pairs
C          must be >= 'nv'
C  local : (i*4)  nout    = parameter = max. no. of output temp/rrc pairs
C          must be >= 'maxt' & 'npspl'
C  local : (i*4)  iarr    = array subscript used for temp/rrc pairs
C  local : (i*4)  iopt    = defines the boundary derivatives for the
C      spline routine 'xxspln', see 'xxspln'.
C      (valid values = <0, 0, 1, 2, 3, 4).
C  local : (l*4)  lsetx   = .true.  => set up spline parameters relating
C         to 'xin' axis.
C      .false. => do not set up spline parameters
C         relating to 'xin' axis.
C         (they were set in a previous call)
C      (value set to .false. by 'xxsple')
C  local : (r*8)  xin()   = log( 'tin()' )
C  local : (r*8)  yin()   = log( 'rrcin()' )
C  local : (r*8)  xout()  = log(temperatures at which splines required)
C  local : (r*8)  yout()  = log(output spline interpolated rrc values)
C  local : (r*8)  df()    = spline interpolated derivatives
C
C
C routines:
C     routine    source brief description
```

```
C     ----------------------------------------------------------
C     xxspln     adas spline subroutine
C     r8fun1     adas real*8 function: ( x -> x )
C
C author:  Paul Bryans (University of Strathclyde)
C
C date:    01/12/04
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             MAXT,        NBLOCK,      NDTEM,       NTMAX
      REAL*8              RRCIN(NTMAX),             RRCOUT(NDTEM)
      REAL*8              TIN(NTMAX),  TOUT(NDTEM)
```

3.76 bdcf3: Subroutine bdcf3 from library adas2xx

```
       subroutine bdcf3( f    , e    , n    , l    , z     ,
     &                   x0   , x1   , h
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** fortran77 routine: bdcf3 ************************
C
C  Purpose:  Tabulates asymptotically decaying bound coulomb function
C
C
C  Subroutine:
C
C  input : (r*8)  e      = energy (Ryd) : must be <0 for a bound state
C  input : (r*8)  n      = principal quantum number
C  input : (r*8)  l      = orbital anular momentum quantum number
C  input : (r*8)  z      = ion charge +1
C  input : (r*8)  x0     = inner turning point of potential
C  input : (r*8)  x1     = outer turning point of potential
C  input : (r*8)  h      = interval for tabulation
C
C  output: (r*8)  f()    = Coulomb function
C                          1st dim: tabulation index
C
C  Routines:
C          none
C
C  Author:  William Osborn (Tessella support services plc)
C
C  Date:     4 July 1996
C
C  Update: MG O'Mullane  19/12/01  Removed junk from > column 72
C
C  Update: HP Summers    21/05/04  made implicit none and restructured
C
C
C  Version  : 1.1                          Date: 04-07-96
C  Modified : William Osborn
C                - First version.
C
C  Version  : 1.2                          Date: 19-12-01
C  Modified : Martin O'Mullane
C                - removed junk from > column 72.
C
C  Version  : 1.3                          Date: 21-15-04
C  Modified : Hugh Summers
C                - Restructured as above.

C-----------------------------------------------------------------------
      INTEGER             L,           N
      REAL*8              E,           F(1000),     H,           X0
      REAL*8              X1,          Z
```

3.77 bdcf7: Subroutine bdcf7 from library adas2xx

```
       subroutine bdcf7(n,l,qd,jsn,z0,nshell,nc,numel,alfa,zl,z1,z2,z3,
     &                  zs,x0,x1,x2,xmax,h,f,c)
C-----------------------------------------------------------------------
C
C  ****************** fortran77 program: bcdf7.for *********************
C
C  Purpose: Calculates a numerical radial wave function in a distorted
C           Coulomb potential described by shell screening.
C
C           The code permits a search for screening parameters given
C           the eigenenergy of for the energy given the screening
C           parameters
C           (original by A. Burgess, DAMTP, University of Cambridge)
C
C
C  Subroutine:
C
C  input : (i*4)  n         = principal quantum number
C  input : (i*4)  l         = orbital quantum number
C  input : (i*4)  qd        = quantum defect for valence electron
C  input : (i*4)  jsn       = -1 => Jucys potential form adopted
C                           = 0  => Slater potential form adopted
C  input : (i*4)  z0        = nuclear charge
C  input : (i*4)  nshell    = number of screening shells
C  input : (i*4)  nc()      = principal quantum number of screening shell
C                             1st dim: index of screening shells
C  input : (i*4)  numel()   = number of electrons in screening shell
C  i/o   : (r*8)  alfa()    = screening parameters
C                             1st dim: screening shell index.
C  input : (r*8)  zl()      =
C  input : (r*8)  z1        =
C  input : (r*8)  z2        =
C  input : (r*8)  z3        =
C  input : (r*8)  zs()      =
C  input : (r*8)  x0        =
C  input : (r*8)  x1        =
C  input : (r*8)  x2        =
C  input : (r*8)  xmax      =
C  input : (r*8)  h         =
C
C  output: (r*8)  f()       = wave function tabulation
C  output: (r*8)  c         =
C
C
C  Routines:
C           routine    source    brief description
C           ------------------------------------------------------------
C           zeffl      adas
C           bdcf3      adas
C           bdcf4      adas
C           fcf6       adas
C           i4unit     adas      fetch unit number for output of messages
C
C
C  Author:  William Osborn (Tessella support services plc)
C
C  Date:    4th july 1996
C
C  Update: M O'Mullane 19-12-01 removed junk from > column 72.
```

```
C
C
C  Update: HP Summers  24/05/04  restructure and addded standard warning
C
C
C  Unix-idl port:
C
C  Version: 1.1                          Date: 04-07-96
C  Modified: William Osborn
C               - first version.
C
C  Version: 1.2                          Date: 19-12-01
C  Modified: Martin O'Mullane
C               - removed junk from > column 72.
C
C  Version: 1.3                          Date: 25-05-2004
C  Modified: H P Summers
C               - restructure.
C
C  Version: 1.4                          Date: 17-05-2007
C  Modified: Allan Whiteford
C               - Updated comments as part of subroutine documentation
C                 procedure.
C
C-----------------------------------------------------------------------
      INTEGER             JSN,          L,            N,            NC(10)
      INTEGER             NSHELL,       NUMEL(10)
      REAL*8              ALFA(10),     C,            F(1000),      H
      REAL*8              QD,           X0,           X1,           X2
      REAL*8              XMAX,         Z0,           Z1,           Z2
      REAL*8              Z3,           ZL(1000),     ZS(100)
```

3.78 bddrsm2: Subroutine bddrsm2 from library adas2xx

```
       SUBROUTINE BDDRSM2( NDREP , NDT  ,
     &                     MAXTM  , IREPMAX, IREP , DRMF , DRMS ,
     &                     EIJN   , PWTEMP    )
       IMPLICIT REAL*8(A-H,O-Z)
C-----------------------------------------------------------------------
C
C *************** FORTRAN 77 SUBROUTINE: BDDRSUM *********************
C
C    VERSION   2.0
C
C    AUTHOR:   WILLIAM J. DICKSON
C              JET JOINT UNDERTAKING
C
C    DATE:     14TH DECEMBER 1992
C
C
C    PURPOSE:
C    --------
C       TO SUM BADNELL DIELECTRONIC RATE COEFFICIENT DATA OVER THE
C       REPRESENTATIVE SET TO GIVE ZERO DENSITY TOTAL AND
C       RADIATED POWER FROM SATELLITE LINES
C
C       CALLING PROGRAM BDMNCLT
C
C
C    INPUT:
C    ------
C     NDREP          -  MAXIMUM NUMBER OF REPRESENTATIVE LEVELS
C     NDT            -  MAXIMUM NUMBER OF TEMPERATURES
C     DRMF(,)        -  BADNELL DIELECTRONIC DATA
C                           1ST INDEX   -  REPRESENTATIVE LEVEL
C                           2ND INDEX   -  TEMPERATURE
C     NBT            -  NO OF TEMPERATURES
C     IREPMAX        -  NO OF REPRESENTATIVE LEVELS
C     IREP           -  SET OF REPRESENTATIVE LEVELS
C     EIJN()         -  SAT. ENERGY AS A FUNCTION OF REPRESENTATIVE LEVEL
C                          UNITS - KELVIN
C    OUTPUT:
C    -------
C     DRMS()         -  SUMMED DR RATE COEFFICIENTS
C                           1ST INDEX    -  TEMPERATURE
C
C     PWTEMP()       -  SAT. RADIATED POWER (UNITS ERG S-1 CM3)
C                           1ST INDEX    -  TEMPERATURE
C
C
C    UPDATE:
C    -------
C     15/12/92   REVISED ALGOROTHM HAS BETTER AGREEMENT WITH
C                INTERNAL SUM CALCULATED BY MAINCL
C-----------------------------------------------------------------------
C
C UNIX-IDL CONVERSION:
C
C VERSION: 1.1                          DATE: 22-08-96
C MODIFIED: WILLIAM OSBORN
C               - FIRST CONVERTED. NO CHANGES.
C
C-----------------------------------------------------------------------
```

```
C-----------------------------------------------------------------------
C
      INTEGER             IREP(NDREP), IREPMAX,     MAXTM,       NDREP
      INTEGER             NDT
      REAL*8              DRMF(NDREP,NDT),          DRMS(NDT)
      REAL*8              EIJN(NDREP), PWTEMP(NDT)
```

3.79 bdwr14: Subroutine bdwr14 from library adas2xx

```
      SUBROUTINE  BDWR14( IUNIT  , NDLEV  , NDJLEV , NDPRT  ,
     &                    NDPRTI , NDMET  , NDT    , NVMAX  , NDTRN ,
     &                    USERID , DATE   ,
     &                    TITLED , IZ     , IZ0    , IZ1    , BWNO  ,
     &                    NPL    , BWNOA  , LBSETA , PRTWTA , CPRTA ,
     &                    NTRM   , NPTRM  , NLVL   , NPLVL  ,
     &                    ISTRM  , IGTRM  , ISPTRM , IGPTRM ,
     &                    ISLVL  , FSLVL  , JTREF  ,
     &                    ISPLVL , FSPLVL , JTPREF ,
     &                    SEQSYM , IGZ    , IGZ0   , IGZ1   ,
     &                    NPRF   , NPRFM  , IPRFM  , NPRI   , IPRI   ,
     &                    IPA    , CSTRPA , ISPA   , ILPA   , XJPA  ,
     &                    WPA    , NGLEV  , BWNI   , NLEVM  , ILEVM  ,
     &                    WGA    , NTE    , TEA    , LRION  , RION   ,
     &                    LSJ    , IL     ,
     &                    IA     , CSTRGA , ISA    , ILA    , XJA   ,
     &                    WA     , CPLA   , NPLA   , IPLA   , ZPLA   ,
     &                    NV     , SCEF   , ITRAN  ,
     &                    TCODE  , I1A    , I2A    , AVAL   , SCOM  ,
     &                    DSNSP  , DSNBD  , DSNXR  , IERROR )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: BDWR14 *********************
C
C  PURPOSE: TO INTERPOLATE IONISATION RATE DATA FROM ADF23 FILE
C           AND WRITE SUPPLEMENTED ADF04 FILE.
C
C  CALLING PROGRAM: ADAS213
C
C  SUBROUTINE:
C
C  INPUT : (I*4) IUNIT       = UNIT TO WHICH INPUT FILE IS ALLOCATED
C
C  INPUT : (I*4) NDLEV       = MAX. NO. OF TERMS THAT CAN BE READ
C  INPUT : (I*4) NDJLEV      = MAX. NO. OF LEVELS THAT CAN BE READ
C  INPUT : (I*4) NDPRT       = MAX. NO. OF PARENT STATES
C  INPUT : (I*4) NDPRTI      = MAX. NO. OF INTERMEDIATE PARENT STATES
C  INPUT : (I*4) NDMET       = MAX. NUMBER OF METASTABLES ALLOWED
C  INPUT : (I*4) NDT         = MAX. NUMBER OF ELECTRON TEMPS IN ADF23 FILE
C  INPUT : (I*4) NVMAX       = MAX. NUMBER OF ELECTRON TEMPS IN ADF04 FILE
C  INPUT : (I*4) NDTRN       = MAX. NUMBER OF TRANSITIONS THAT CAN BE READ
C  INPUT : (C*10)USERID      = USER IDENTIFIER OF CODE EXECUTOR.
C  INPUT : (C*8) DATE        = DATE (AS DD/MM/YY).
C  INPUT : (C*3) TITLED      = ELEMENT SYMBOL.
C  INPUT : (I*4) IZ          =  RECOMBINED ION CHARGE READ
C  INPUT : (I*4) IZ0         =         NUCLEAR CHARGE READ
C  INPUT : (I*4) IZ1         = RECOMBINING ION CHARGE READ
C                              (NOTE: IZ1 SHOULD EQUAL IZ+1)
C  INPUT : (R*8) BWNO        = IONISATION POTENTIAL (CM-1) OF LOWEST PARENT
C  INPUT : (I*4) NPL         = NUMBER OF PARENTS ON FIRST LINE AND USED
C                              IN LEVEL ASSIGNMENTS
C  INPUT : (R*8) BWNOA()     = IONISATION POTENTIAL (CM-1) OF PARENTS
C  INPUT : (L*4) LBSETA()    = .TRUE.  - PARENT WEIGHT SET FOR BWNOA()
C                              .FALSE. - PARENT WEIGHT NOT SET FOR BWNOA()
C  INPUT : (R*8) PRTWTA()    = PARENT WEIGHT FOR BWNOA()
C  INPUT : (C*9) CPRTA()     = PARENT NAME IN BRACKETS
C
C  INPUT : (I*4) NTRM        = NUMBER OF TERMS IN X-REF FILE.
C  INPUT : (I*4) NPTRM       = NUMBER OF PARENT TERMS IN X-REF FILE.
```

```
C  INPUT : (I*4) NLVL        = NUMBER OF LEVELS IN X-REF FILE.
C  INPUT : (I*4) NPLVL       = NUMBER OF PARENT LEVELS IN X-REF FILE.
C
C  INPUT : (I*4) ISTRM()     = SPEC. ION FILE TERM INDEX FROM X-REF
C                              1ST.DIM.: TERM COUNTER IN X-REF FILE
C  INPUT : (I*4) IGTRM()     = IONIS. FILE TERM INDEX FROM X-REF
C                              1ST.DIM.: TERM COUNTER IN X-REF FILE
C  INPUT : (I*4) ISPTRM()    = SPEC. ION FILE PRNT. TERM INDEX FROM X-REF
C                              1ST.DIM.: PRNT. TERM COUNTER IN X-REF FILE
C  INPUT : (I*4) IGPTRM()    = IONIS. FILE PRNT. TERM INDEX FROM X-REF
C                              1ST.DIM.: PRNT. TERM COUNTER IN X-REF FILE
C  INPUT : (I*4) ISLVL()     = SPEC. ION FILE LEVEL INDEX FROM X-REF
C                              1ST.DIM.: LEVEL COUNTER IN X-REF FILE
C  INPUT : (R*8) FSLVL()     = FRACTIONATION OF TERM RATES AMONG LEVELS
C                              1ST.DIM.: LEVEL COUNTER IN X-REF FILE
C  INPUT : (I*4) JTREF()     = SP. ION FILE TERM ASSOCIATED WITH LEVEL
C                              FROM X-REF FILE.
C                              1ST.DIM.: LEVEL COUNTER IN X-REF FILE
C  INPUT : (I*4) ISPLVL()    = SPEC. ION FILE PRNT. LEVEL INDEX FROM X-REF
C                              1ST.DIM.: PRNT. LEVEL COUNTER IN X-REF FILE
C  INPUT : (R*8) FSPLVL()    = FRACTIONATION OF PRNT. TERM RATES AMONG
C                              PRNT. LEVELS
C                              1ST.DIM.: LEVEL COUNTER IN X-REF FILE
C  INPUT : (I*4) JTPREF()    = SP. ION FILE PRNT. TERM ASSOCIATED WITH
C                              PRNT. LEVEL FROM X-REF FILE.
C                              1ST.DIM.: LEVEL COUNTER IN X-REF FILE
C  INPUT : (C*2)  SEQSYM     = RECOMBINED ION SEQ
C  INPUT : (I*4)  IGZ        = RECOMBINED ION CHARGE FROM ADF23 FILE
C  INPUT : (I*4)  IGZ0       = NUCLEAR CHARGE FROM ADF23 FILE
C  INPUT : (I*4)  IGZ1       = RECOMBINING ION CHARGE FROM ADF23 FILE
C  INPUT : (I*4)  NPRF       = NUMBER OF FINAL PARENTS
C  INPUT : (I*4)  NPRFM      = NUMBER OF FINAL PARENTS WHICH ARE METASTABLES
C  INPUT : (I*4)  IPRFM()    = CROSS-REFERENCING OF FINAL METASTABLE
C                              PARENTS TO FINAL PARENT LIST.
C  INPUT : (I*4)  NPRI       = NUMBER OF FINAL PARENTS WHICH ARE INTERMEDIATE
C                              PARENTS FOR REPR. N-SHELL DOUBLY EXCITED STATES
C  INPUT : (I*4)  IPRI()     = CROSS-REFERENCING OF INTERMEDIATE
C                              PARENTS TO FINAL PARENT LIST.
C  INPUT : (I*4)  IPA()      = INDEX OF FINAL PARENT ENERGY LEVELS
C  INPUT : (C*18) CSTRPA()   = NOMENCL./CONFIG. FOR PARENT LEVEL 'IPA()'
C  INPUT : (I*4)  ISPA()     = MULTIPLICITY FOR PARENT LEVEL 'IPA()'
C                              NOTE: (ISPA-1)/2 = QUANTUM NUMBER (SP)
C  INPUT : (I*4)  ILPA()     = QUANTUM NUMBER (LP) FOR PARENT LEVEL 'IPA()'
C  INPUT : (R*8)  XJPA()     = QUANTUM NUMBER (JP) FOR PARENT LEVEL 'IPA()'
C                              NOTE: (2*XJPA)+1 = STATISTICAL WEIGHT
C  INPUT : (R*8)  WPA()      = ENERGY RELATIVE TO PARENT LEVEL 1 (CM-1)
C                              FOR PARENT LEVEL 'IPA()'
C  INPUT : (I*4)  NGLEV      = NUMBER OF ENERGY LEVELS (TERMS) OF THE
C                              IONISING ION FROM ADF23 FILE
C  INPUT : (R*8)  BWNI       = IONISATION POTENTIAL (CM-1) OF LOWEST LEVEL
C                              OF IONISING ION
C  INPUT : (I*4)  NLEVM      = NUMBER OF IONISING ION LEVELS WHICH ARE
C                              METASTABLES
C  INPUT : (I*4)  ILEVM()    = CROSS-REFERENCING OF IONISNG ION METASTABLES
C                              TO IONISING ION LEVEL LIST.
C  INPUT : (R*8)  WGA()      = ENERGY RELATIVE TO RECOMBINED LEVEL 1 (CM-1)
C                              FOR RECOMBINED LEVEL 'IA()' FROM ADF23 FILE
C  INPUT : (I*4)  NTE        = NUMBER OF ELECTRON TEMPS. FROM ADF23 FILE
C  INPUT : (R*8)  TEA()      = ELECTRON TEMPERATURES (K) FROM ADF23 FILE
C  INPUT : (L*4)  LRION(,)   = .TRUE.  => DATA PRESENT FOR FINAL STATE
C                              .FALSE. => DATA NOT PRESENT FOR FINAL STATE
```

```
C                                1ST.DIM: IONISING ION METASTABLE INDEX
C                                2ND.DIM: FINAL PARENT INDEX
C  INPUT : (R*8)  RION(,,)    = STATE SELECTIVE DIRECT IONISATION COEFFICIENTS
C                                1ST.DIM: IONISING ION METASTABLE INDEX
C                                2ND.DIM: FINAL PARENT INDEX
C                                3RD.DIM: ELECTRON TEMPERATURE INDEX
C  INPUT : (L*4)  LSJ           = .TRUE. => J-RESOL. INFO. IN X-REF FILE
C                                .FALSE.=> NO J-RESOL. IN X-REF FILE
C  INPUT : (I*4)  IL            = INPUT DATA FILE: NUMBER OF ENERGY LEVELS
C
C  INPUT : (I*4)  IA()          = ENERGY LEVEL INDEX NUMBER
C  INPUT : (C*18) CSTRGA()      = NOMENCLATURE/CONFIGURATION FOR LEVEL 'IA()'
C  INPUT : (I*4)  ISA()         = MULTIPLICITY FOR LEVEL 'IA()'
C                                NOTE: (ISA-1)/2 = QUANTUM NUMBER (S)
C  INPUT : (I*4)  ILA()         = QUANTUM NUMBER (L) FOR LEVEL 'IA()'
C  INPUT : (R*8)  XJA()         = QUANTUM NUMBER (J-VALUE) FOR LEVEL 'IA()'
C                                NOTE: (2*XJA)+1 = STATISTICAL WEIGHT
C  INPUT : (R*8)  WA()          = ENERGY RELATIVE TO LEVEL 1 (CM-1) FOR LEVEL
C                                'IA()'
C  INPUT : (C*1)  CPLA()        = CHAR. SPECIFYING 1ST PARENT FOR LEVEL 'IA()'
C                                INTEGER - PARENT IN BWNOA() LIST
C                                'BLANK' - PARENT BWNOA(1)
C                                  'X'   - DO NOT ASSIGN A PARENT
C                                1ST DIM.: LEVEL INDEX
C  INPUT : (I*4)  NPLA()        = NO. OF PARENT/ZETA CONTRIBUTIONS TO IONIS.
C                                OF LEVEL
C                                1ST DIM.: PARENT INDEX
C  INPUT : (I*4)  IPLA(,)       = PARENT INDEX FOR CONTRIBUTIONS TO IONIS.
C                                OF LEVEL
C                                1ST DIM.: PARENT INDEX
C                                2ND DIM.: LEVEL INDEX
C  INPUT : (R*8)  ZPLA(,        = EFF. ZETA PARAM. FOR CONTRIBUTIONS TO IONIS.
C                                OF LEVEL
C                                1ST DIM.: PARENT INDEX
C  INPUT : (C*92) CIONP         = STRING CONTAINING LEVEL TERMINATOR AND
C                                IONISATION POTENTIALS
C
C  INPUT : (I*4)  NV            = INPUT DATA FILE: NUMBER OF GAMMA/TEMPERATURE
C                                PAIRS FOR A GIVEN TRANSITION.
C  INPUT : (R*8)  SCEF()        = INPUT DATA FILE: ELECTRON TEMPERATURES (K)
C                                (INITIALLY JUST THE MANTISSA. SEE 'ITPOW()')
C                                (NOTE: TE=TP=TH IS ASSUMED)
C
C  INPUT : (I*4)  ITRAN         = INPUT DATA FILE: NUMBER OF TRANSITIONS
C  INPUT : (C*1)  TCODE()       = TRANSITION: DATA TYPE POINTER:
C                                ' ' => Electron Impact   Transition
C                                'P' => Proton   Impact   Transition
C                                'H' => Charge   Exchange Recombination
C                                'R' => Free     Electron Recombination
C  INPUT : (I*4)  I1A()         = TRANSITION:
C                                 LOWER ENERGY LEVEL INDEX (CASE ' ' & 'P')
C                                 SIGNED PARENT      INDEX (CASE 'H' & 'R')
C  INPUT : (I*4)  I2A()         = TRANSITION:
C                                 UPPER ENERGY LEVEL INDEX (CASE ' ' & 'P')
C                                 CAPTURING    LEVEL INDEX (CASE 'H' & 'R')
C  INPUT : (R*8)  AVAL()        = TRANSITION:
C                                 A-VALUE (SEC-1)          (CASE ' ')
C                                 NEUTRAL BEAM ENERGY      (CASE 'H')
C                                 NOT USED                 (CASE 'P' & 'R')
C  INPUT : (R*8)  SCOM(,)       = TRANSITION:
C                                 GAMMA VALUES             (CASE ' ' & 'P')
```

```
C                              RATE COEFFT. (CM3 SEC-1) (CASE 'H' & 'R')
C                             1ST DIMENSION - TEMPERATURE 'SCEF()'
C                             2ND DIMENSION - TRANSITION NUMBER
C  INPUT : (C*80) DSNSP      = INPUT ADF04 FILE NAME
C  INPUT : (C*80) DSNBD      = ADF23 IONISATION DATA FILE NAME
C  INPUT : (C*80) DSNXR      = ADF18 CROSS-REFERENCE FILE NAME
C  OUTPUT: (I*4)  IERROR     = 0 => X-REF FILE OK
C                              1 => FAULT IN XREF FILE DATASETS
C                              2 => FAULT IN XREF FILE TERM COUNT
C                              3 => FAULT IN XREF FILE LEVEL COUNT
C
C
C  ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          XXSPLN     ADAS      SPLINE CURVE FITTING/INTERPOLATION.
C
C  AUTHOR:  H. P. SUMMERS, UNIVERSITY OF STRATHCLYDE
C          JA8.08
C          TEL. 0141-553-4196
C
C  DATE:    03/04/98
C
C  UPDATE:
C
C  VERSION: 1.1 DATE: 23-06-98
C  MODIFIED: HUGH SUMMERS
C FIRST VERSION.
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      CHARACTER           CPLA(NDLEV)
      CHARACTER*9         CPRTA(NDMET)
      CHARACTER*18        CSTRGA(NDLEV),            CSTRPA(NDPRT)
      CHARACTER*8         DATE
      CHARACTER*80        DSNBD,       DSNSP,       DSNXR
      CHARACTER*2         SEQSYM
      CHARACTER           TCODE(NDTRN)
      CHARACTER*3         TITLED
      CHARACTER*10        USERID
      INTEGER             I1A(NDTRN),  I2A(NDTRN),  IA(NDLEV),   IERROR
      INTEGER             IGPTRM(NDPRT),            IGTRM(NDLEV)
      INTEGER             IGZ,         IGZ0,        IGZ1,        IL
      INTEGER             ILA(NDLEV),  ILEVM(NDLEV)
      INTEGER             ILPA(NDPRT), IPA(NDPRT),  IPLA(NDMET,NDLEV)
      INTEGER             IPRFM(NDPRT),             IPRI(NDPRTI)
      INTEGER             ISA(NDLEV),  ISLVL(NDJLEV)
      INTEGER             ISPA(NDPRT), ISPLVL(NDPRT)
      INTEGER             ISPTRM(NDPRT),            ISTRM(NDLEV)
      INTEGER             ITRAN,       IUNIT,       IZ,          IZ0
      INTEGER             IZ1,         JTPREF(NDPRT)
      INTEGER             JTREF(NDJLEV),            NDJLEV,      NDLEV
      INTEGER             NDMET,       NDPRT,       NDPRTI,      NDT
      INTEGER             NDTRN,       NGLEV,       NLEVM,       NLVL
      INTEGER             NPL,         NPLA(NDLEV), NPLVL,       NPRF
      INTEGER             NPRFM,       NPRI,        NPTRM,       NTE
      INTEGER             NTRM,        NV,          NVMAX
      LOGICAL             LBSETA(NDMET),            LRION(NDMET,NDPRT)
      LOGICAL             LSJ
      REAL*8              AVAL(NDTRN), BWNI,        BWNO
      REAL*8              BWNOA(NDMET),             FSLVL(NDJLEV)
```

```
REAL*8              FSPLVL(NDPRT),              PRTWTA(NDMET)
REAL*8              RION(NDMET,NDPRT,NDT),      SCEF(NVMAX)
REAL*8              SCOM(NVMAX,NDTRN),          TEA(NDT)
REAL*8              WA(NDLEV),    WGA(NDLEV),   WPA(NDPRT)
REAL*8              XJA(NDLEV),   XJPA(NDPRT),  ZPLA(NDMET,NDLEV)
```

3.80 bdxref: Subroutine bdxref from library adas2xx

```
      SUBROUTINE  BDXREF( IUNIT  , NDLEV  , NDJLEV , NDPRT  ,
     &                    DSNSP  , DSNBD  , DSNSPO ,
     &                    NTRM   , NPTRM  , NLVL   , NPLVL  ,
     &                    ISTRM  , IGTRM  , ISPTRM , IGPTRM ,
     &                    ISLVL  , FSLVL  , JTREF  ,
     &                    ISPLVL , FSPLVL , JTPREF ,
     &                    LSJ    , IERROR
     &                  )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: BDXREF *********************
C
C  PURPOSE: TO FETCH DATA FROM INPUT ADF18/A23_A04 CROSS-REFERENCE FILE.
C
C  CALLING PROGRAM: ADAS213
C
C
C  SUBROUTINE:
C
C  INPUT : (I*4) IUNIT       = UNIT TO WHICH INPUT FILE IS ALLOCATED
C
C  INPUT : (I*4) NDLEV       = MAX. NO. OF TERMS THAT CAN BE READ
C  INPUT : (I*4) NDJLEV      = MAX. NO. OF LEVELS THAT CAN BE READ
C  INPUT : (I*4) NDPRT       = MAX. NO. OF PARENT STATES
C
C  OUTPUT: (C*80) DSNSP      = INPUT ADF04 FILE NAME
C  OUTPUT: (C*80) DSNBD      = ADF23 IONISATION DATA FILE NAME
C  OUTPUT: (C*80) DSNSP      = OUTPUT ADF04 FILE NAME
C
C  OUTPUT: (I*4) NTRM        = NUMBER OF TERMS IN X-REF FILE.
C  OUTPUT: (I*4) NPTRM       = NUMBER OF PARENT TERMS IN X-REF FILE.
C  OUTPUT: (I*4) NLVL        = NUMBER OF LEVELS IN X-REF FILE.
C  OUTPUT: (I*4) NPLVL       = NUMBER OF PARENT LEVELS IN X-REF FILE.
C
C  OUTPUT: (I*4) ISTRM()     = SPEC. ION FILE TERM INDEX FROM X-REF
C                              1ST.DIM.: TERM COUNTER IN X-REF FILE
C  OUTPUT: (I*4) IGTRM()     = IONIS. FILE TERM INDEX FROM X-REF
C                              1ST.DIM.: TERM COUNTER IN X-REF FILE
C  OUTPUT: (I*4) ISPTRM()    = SPEC. ION FILE PRNT. TERM INDEX FROM X-REF
C                              1ST.DIM.: PRNT. TERM COUNTER IN X-REF FILE
C  OUTPUT: (I*4) IGPTRM()    = IONIS. FILE PRNT. TERM INDEX FROM X-REF
C                              1ST.DIM.: PRNT. TERM COUNTER IN X-REF FILE
C  OUTPUT: (I*4) ISLVL()     = SPEC. ION FILE LEVEL INDEX FROM X-REF
C                              1ST.DIM.: LEVEL COUNTER IN X-REF FILE
C  OUTPUT: (I*4) JTREF()     = SP. ION FILE TERM ASSOCIATED WITH LEVEL
C                              FROM X-REF FILE.
C                              1ST.DIM.: LEVEL COUNTER IN X-REF FILE
C  OUTPUT: (I*4) ISPLVL()    = SPEC. ION FILE PRNT. LEVEL INDEX FROM X-REF
C                              1ST.DIM.: PRNT. LEVEL COUNTER IN X-REF FILE
C  OUTPUT: (I*4) JTPREF()    = SP. ION FILE PRNT. TERM ASSOCIATED WITH
C                              PRNT. LEVEL FROM X-REF FILE.
C                              1ST.DIM.: LEVEL COUNTER IN X-REF FILE
C  OUTPUT: (I*4) ISPTRM()    = SPEC. ION FILE PRNT. TERM INDEX FROM X-REF
C                              1ST.DIM.: PRNT. TERM COUNTER IN X-REF FILE
C  OUTPUT: (I*4) IGPTRM()    = IONIS. FILE PRNT. TERM INDEX FROM X-REF
C                              1ST.DIM.: PRNT. TERM COUNTER IN X-REF FILE
C
C  OUTPUT: (R*8) FSLVL()     = FRACTIONATION OF TERM RATES AMONG LEVELS
C                              1ST.DIM.: LEVEL COUNTER IN X-REF FILE
```

```
C  OUTPUT: (R*8) FSPLVL()    = FRACTIONATION OF PRNT. TERM RATES AMONG
C                             PRNT. LEVELS
C                             1ST.DIM.: LEVEL COUNTER IN X-REF FILE
C  OUTPUT: (L*4) LSJ         = .TRUE. => J-RESOL. INFO. IN X-REF FILE
C                             .FALSE.=> NO J-RESOL. IN X-REF FILE
C  OUTPUT: (I*4) IERROR      = 0 => X-REF FILE OK
C                             1 => FAULT IN XREF FILE DATASETS
C                             2 => FAULT IN XREF FILE TERM COUNT
C                             3 => FAULT IN XREF FILE LEVEL COUNT
C
C          (C*18) C18     = GENERAL CHARACTER STRING
C
C  ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4UNIT     ADAS      FETCH UNIT NUMBER FOR OUTPUT OF MESSAGES
C          XXFLNM     ADAS      CONVERT SHORT FILE NAME TO FULL NAME
C
C  AUTHOR:  H. P. SUMMERS, UNIVERSITY OF STRATHCLYDE
C          JA8.08
C          TEL. 0141-553-4196
C
C  DATE:    02/04/98
C
C  UPDATE:
C
C  VERSION: 1.1                                        DATE: 23-06-98
C  MODIFIED: HUGH SUMMERS
C               FIRST VERSION.
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      CHARACTER*80        DSNBD,       DSNSP,       DSNSPO
      INTEGER             IERROR,      IGPTRM(NDPRT)
      INTEGER             IGTRM(NDLEV),             ISLVL(NDJLEV)
      INTEGER             ISPLVL(NDPRT),            ISPTRM(NDPRT)
      INTEGER             ISTRM(NDLEV),             IUNIT
      INTEGER             JTPREF(NDPRT),            JTREF(NDJLEV)
      INTEGER             NDJLEV,      NDLEV,       NDPRT,       NLVL
      INTEGER             NPLVL,       NPTRM,       NTRM
      LOGICAL             LSJ
      REAL*8              FSLVL(NDJLEV),            FSPLVL(NDPRT)
```

3.81 bexcoef: Subroutine bexcoef from library adas2xx

```
       SUBROUTINE BEXCOEF ( FILELS , IFAIL  , LFXIST ,
     &                      NION   , MAXT   ,
     &                      INDA   , NIND   , NSPEC  ,
     &                      LPSEL  , LZSEL  , LISEL  , LHSEL  , LRSEL  ,
     &                      TEVA   , TPVA   , THVA   ,
     &                      DENSA  , DENSPA , RATHA  , RATIA  ,
     &                      ZEFF   ,
     &                      COEF   , SPEC   , POPAR
     &                    )
C-----------------------------------------------------------------------
C  ****************** FORTRAN77 SUBROUTINE: BEXCOEF *********************
C
C  VERSION:  1.1
C
C  CALLING PROGRAM: ADAS214
C
C  PURPOSE:  TO  CALCULATE  COMPLETE SETS OF SPECTRUM LINE EMISSIVITIES
C            FOR THE IONS OF AN ELEMENT
C
C            PROCESSES  CAN  INCLUDE  ELECTRON AND PROTON IMPACT,  SPON-
C            TANEOUS EMISSION,   FREE ELECTRON RECOMBINATION  AND CHARGE
C            EXCHANGE RECOMBINATION DEPENDING ON THE INPUT DATA SET.
C
C            ACCEPTS MULTIPLE INPUT FILES.  DESIGNED FOR USE IN G(T)
C            CALCULATIONS ETC.
C
C
C  DATA:     THE SOURCE DATA ARE SPECIFIC ION EXCITATION FILES STORED AS
C            PARTITIONED DATA SET MEMBERS AS FOLLOWS:-
C
C                    'JETSHP.<SE>LIKE.DATA(<MEMBER>)'
C
C            ACCORDING TO ADAS DATA FORMAT ADF04.
C
C
C  INPUT  :(C*60) FILELS()= INPUT COPASE FILE NAMES
C          (L*4)  LFXIST()= .TRUE.  => COPASE FILE FOR THIS ION
C                           .FALSE. => NO COPASE FILE FOR THIS ION
C          (I*4)  NION    = NUMBER OF IONS TO BE COMPUTED
C          (I*4)  MAXT    = NUMBER OF TEMPERATURE/DENSITY PAIRS
C          (L*4)  LPSEL   = .TRUE.  => PROTON DATA TO BE INCLUDED
C                           = .FALSE. => PROTON DATA TO BE EXCLUDED
C          (L*4)  LZSEL   = .TRUE.  => SCALE PROTON DATA WITH ZEFF
C                           = .FALSE. => DO NOT SCALE PROTON DATA
C          (L*4)  LISEL   = .TRUE.  => IONISATION TO BE INCLUDED
C                           = .FALSE. => IONISATION TO BE EXCLUDED
C          (L*4)  LHSEL   = .TRUE.  => CHARGE TRANSFER TO BE INCLUDED
C                           = .FALSE. => CHARGE TRANSFER TO BE EXCLUDED
C          (L*4)  LRSEL   = .TRUE.  => RECOMBINATION TO BE INCLUDED
C                           = .FALSE. => RECOMBINATION TO BE EXCLUDED
C          (R*8)  TEVA()  = ELECTRON TEMPERATURES (EV)
C          (R*8)  TPVA()  = PROTON TEMPERATURES (EV)
C          (R*8)  THVA()  = NEUTRAL HYDROGEN TEMPERATURES (EV)
C          (R*8)  DENSA() = ELECTRON DENSITIES  (CM-3)
C          (R*8)  DENSPA()= PROTON DENSITIES  (CM-3)
C          (R*8)  RATHA() = RATIO (NEUTRAL H DENSITY/ELECTRON DENSITY)
C          (R*8)  RATIA() = RATIO (N(Z+1)/N(Z) STAGE ABUNDANCES)
C
C  OUTPUT :(I*4)  IFAIL   = 0    SUBROUTINE SUCCESSFUL
```

```
C                            1    SUBROUTINE FAILURE OR WARNING
C          (I*4)  INDA(,) = IDENTIFIER FOR SPECTRUM LINE (10000*IL+IU)
C                           1ST DIMENSION - INDEX OF LINES FOR AN ION
C                           2ND DIMENSION - ION COUNT INDEX
C          (I*4)  NIND()  = NUMBER OF LINES FOR AN ION
C                           1ST DIMENSION - ION COUNT INDEX
C          (I*4)  NSPEC() = NUMBER OF LEVELS FOR AN ION
C                           1ST DIMENSION - ION COUNT INDEX
C          (R*8)  ZEFF    = PLASMA Z EFFECTIVE ( IF 'LZSEL' = .TRUE.)
C                             (IF 'LZSEL' = .FALSE. => 'ZEFF=1.0')
C          (R*8)  COEF(,,)= EMISSIVITY FOR SPECTRUM LINE (10000*J+I)
C                           1ST DIMENSION - INDEX OF LINES FOR AN ION
C                           2ND DIMENSION - TEMPERATURE INDEX
C                           3RD DIMENSION - ION COUNT INDEX
C          (C*51) SPEC(,) = INFORMATION STRING FOR LEVEL
C                           1ST DIMENSION - INDEX OF LEVELS FOR AN ION
C                           2ND DIMENSION - ION COUNT INDEX
C
C
C  PROGRAM:
C          (I*4)  NDLEV   = PARAMETER = MAX. NUMBER OF LEVELS ALLOWED
C          (I*4)  NDTRN   = PARAMETER = MAX. NO. OF TRANSITIONS ALLOWED
C          (I*4)  NDTEM   = PARAMETER = MAX. NO. OF TEMPERATURES ALLOWED
C          (I*4)  NZDIM   = PARAMETER = MAX. NO. OF IONS ALLOWED
C          (I*4)  NDMET   = PARAMETER = MAX. NO. OF METASTABLES ALLOWED
C
C          (I*4)  IUNT10  = PARAMETER = INPUT UNIT FOR COPASE DATA SET
C                                       PASSING FILE.
C          (I*4)  L1      = PARAMETER = 1
C
C          (R*8)  D1      = PARAMETER = 1.0D0
C
C          (I*4)  ICNTE   = NUMBER OF ELECTRON IMPACT TRANSITIONS INPUT
C          (I*4)  ICNTP   = NUMBER OF PROTON IMPACT TRANSITIONS INPUT
C          (I*4)  ICNTR   = NUMBER OF FREE ELECTRON RECOMBINATIONS INPUT
C          (I*4)  ICNTH   = NO. OF CHARGE EXCHANGE RECOMBINATIONS INPUT
C          (I*4)  IL      = INPUT DATA FILE: NUMBER OF ENERGY LEVELS
C          (I*4)  ITRAN   = INPUT DATA FILE: NUMBER OF TRANSITIONS
C          (I*4)  IZ0     = NUCLEAR CHARGE
C          (I*4)  IZ      =  RECOMBINED ION CHARGE
C          (I*4)  IZ1     = RECOMBINING ION CHARGE
C                           (NOTE: IZ1 SHOULD EQUAL IZ+1)
C          (I*4)  MAXLEV  = HIGHEST INDEX LEVEL IN READ TRANSITIONS
C          (I*4)  MAXT    = NO. OF INPUT TEMP/DENS PAIRS ( 1 -> 'NDTEM')
C          (I*4)  NMET    = NUMBER OF METASTABLES (1 <= NMET <= 'NDMET')
C          (I*4)  NORD    = NUMBER OF ORDINARY LEVELS ('IL' - 'NMET')
C          (I*4)  NV      = INPUT DATA FILE: NUMBER OF GAMMA/TEMPERATURE
C                           PAIRS FOR A GIVEN TRANSITION.
C          (I*4)  I       = GENERAL INDEX
C          (I*4)  IT      = TEMPERATURE ARRAY INDEX
C          (I*4)  IS      = ENERGY LEVEL ARRAY INDEX
C
C          (R*8)  TEA()   = INPUT ELECTRON TEMPERATURES (K)
C          (R*8)  TPA()   = INPUT PROTON TEMPERATURES (K)
C          (R*8)  THA()   = INPUT NEUTRAL HYDROGEN TEMPERATURES (K)
C          (R*8)  R8FBCH  = FUNCTION (SEE ROUTINE SECTION BELOW)
C          (R*8)  BWNO    = IONISATION POTENTIAL (CM-1)
C          (R*8)  ZEFFSQ  = 'ZEFF' * 'ZEFF'
C          (R*8)  DMINT   = +1 or -1 DEPENDING ON WHETHER THE  NUMBER OF
C                           ROW   INTERCHANGES   WAS   EVEN   OR    ODD,
C                           RESPECTIVELY, WHEN INVERTING A MATRIX  USING
```

```
C                          'XXMINV'.
C
C          (L*4)  LSOLVE  = .TRUE.  => SOLVE LINEAR EQUATION USING
C                                      'XXMINV'.
C                          .FALSE. =>DO NOT SOLVE LINEAR EQUATION USING
C                                    'XXMINV' - INVERT MATRIX ONLY.
C          (L*4)  OPEN10  = .TRUE.  => FILE ALLOCATED TO UNIT 10.
C                         = .FALSE. => NO FILE ALLOCATED TO UNIT 10.
C          (C*3)  TITLED  = ELEMENT SYMBOL.
C          (C*8)  DATE    = CURRENT DATE AS 'DD/MM/YY'
C          (C*60) DSNINC  = INPUT COPASE DATA SET NAME (MVS DSN)
C          (C*51) CLINE   = LEVEL SPECIFICATION LINE
C
C          (I*4)  IA()    = ENERGY LEVEL INDEX NUMBER
C          (I*4)  ILA()   = QUANTUM NUMBER (L) FOR LEVEL 'IA()'
C          (I*4)  ISA()   = MULTIPLICITY FOR LEVEL 'IA()'
C                           NOTE: (ISA-1)/2 = QUANTUM NUMBER (S)
C          (I*4)  IMETR() = INDEX OF METASTABLE IN COMPLETE LEVEL LIST
C                           (ARRAY SIZE = 'NDMET' )
C          (I*4)  IORDR() = INDEX OF ORDINARY EXCITED LEVELS IN COMPLETE
C                           LEVEL LIST.
C          (I*4)  I1A()   = TRANSITION:
C                            LOWER ENERGY LEVEL INDEX (CASE ' ' & 'P')
C                            NOT USED                 (CASE 'H' & 'R')
C          (I*4)  I2A()   = TRANSITION:
C                            UPPER ENERGY LEVEL INDEX (CASE ' ' & 'P')
C                            CAPTURING    LEVEL INDEX (CASE 'H' & 'R')
C          (I*4)  IETRN() = ELECTRON IMPACT TRANSITION:
C                           INDEX VALUES IN MAIN TRANSITION ARRAYS WHICH
C                           REPRESENT ELECTRON IMPACT TRANSITIONS.
C          (I*4)  IPTRN() = PROTON IMPACT TRANSITION:
C                           INDEX VALUES IN MAIN TRANSITION ARRAYS WHICH
C                           REPRESENT PROTON IMPACT TRANSITIONS.
C          (I*4)  IRTRN() = FREE ELECTRON RECOMBINATION:
C                           INDEX VALUES IN MAIN TRANSITION ARRAYS WHICH
C                           REPRESENT FREE ELECTRON RECOMBINATIONS.
C          (I*4)  IHTRN() = CHARGE EXCHANGE RECOMBINATION:
C                           INDEX VALUES IN MAIN TRANSITION ARRAYS WHICH
C                           REPRESENT CHARGE EXCHANGE RECOMBINATIONS.
C          (I*4)  IE1A()  = ELECTRON IMPACT TRANSITION:
C                            LOWER ENERGY LEVEL INDEX
C          (I*4)  IE2A()  = ELECTRON IMPACT TRANSITION:
C                            UPPER ENERGY LEVEL INDEX
C          (I*4)  IP1A()  = PROTON IMPACT TRANSITION:
C                            LOWER ENERGY LEVEL INDEX
C          (I*4)  IP2A()  = PROTON IMPACT TRANSITION:
C                            UPPER ENERGY LEVEL INDEX
C
C          (R*8)  PAR(,)  =
C          (R*8)  ER()    = ENERGY RELATIVE TO LEVEL 1 (RYDBERGS)
C                           DIMENSION: LEVEL INDEX
C          (R*8)  XIA()   = ENERGY RELATIVE TO ION. POT. (RYDBERGS)
C                           DIMENSION: LEVEL INDEX
C          (R*8)  AA()    = ELECTRON IMPACT TRANSITION: A-VALUE (SEC-1)
C          (R*8)  AVAL()  = TRANSITION:
C                            A-VALUE (SEC-1)          (CASE ' ')
C                            NEUTRAL BEAM ENERGY      (CASE 'H')
C                            NOT USED                 (CASE 'P' & 'R')
C          (R*8)  SCOM(,) = TRANSITION:
C                            GAMMA VALUES             (CASE ' ' & 'P')
C                            RATE COEFFT. (CM3 SEC-1) (CASE 'H' & 'R')
```

```
C                          1ST DIMENSION - TEMPERATURE 'SCEF()'
C                          2ND DIMENSION - TRANSITION NUMBER
C          (R*8)  SCEF()  = INPUT DATA FILE: ELECTRON TEMPERATURES (K)
C          (R*8)  WA()    = ENERGY RELATIVE TO LEVEL 1 (CM-1)
C                           DIMENSION: LEVEL INDEX
C          (R*8)  XJA()   = QUANTUM NUMBER (J-VALUE) FOR LEVEL 'IA()'
C                           NOTE: (2*XJA)+1 = STATISTICAL WEIGHT
C          (R*8)  RHS()   = USED ONLY IF 'LSOLVE=.TRUE.' WHEN CALLING
C                           THE SUBROUTINE 'XXMINV'. CONTAINS THE SET
C                           OF 'N' LINEAR EQUATIONS TO BE SOLVED.
C                           INPUT  TO   'XXMINV': RIGHT HAND SIDE VECTOR
C                           OUTPUT FROM 'XXMINV': SOLUTION VECTOR
C                           (ACTS ONLY AS A DUMMY IN THIS PROGRAM)
C          (R*8)  CIE()   = IONISATION RATE COEFFICIENT VECTOR FOR
C                           FIXED TEMPERATURE.
C                            DIMENSION: ENERGY LEVEL INDEX
C          (R*8)  VHRED() = CHARGE EXCHANGE RECOMBINATION:
C                           VECTOR  OF RECOMBINATION RATE CONTRIBUTIONS
C                           FOR EACH METASTABLE LEVEL.
C                           (UNITS: SEC-1)
C                           VALUES FOR GIVEN TEMPERATURE AND DENSITY.
C                            DIMENSION: METASTABLE LEVEL INDEX
C          (R*8)  VRRED() = FREE ELECTRON RECOMBINATION:
C                           VECTOR  OF RECOMBINATION RATE CONTRIBUTIONS
C                           FOR EACH METASTABLE LEVEL.
C                           (UNITS: SEC-1)
C                           VALUES FOR GIVEN TEMPERATURE AND DENSITY.
C                            DIMENSION: METASTABLE LEVEL INDEX
C
C          (R*8)  EXCRE(,) = ELECTRON IMPACT TRANSITION:
C                            EXCITATION RATE COEFFS (cm**3/s)
C                            PRE  'BXRATE': UNIT GAMMA VALUES
C                            POST 'BXRATE': TRUE VALUES
C                            1st DIMENSION: TEMPERATURE INDEX ('TOUT')
C                            2nd DIMENSION: TRANSITION INDEX
C          (R*8)  DEXCRE(,)= ELECTRON IMPACT TRANSITION:
C                            DE-EXCITATION RATE COEFFS (cm**3/s)
C                            PRE  'BXRATE': UNIT GAMMA VALUES
C                            POST 'BXRATE': TRUE VALUES
C                            1st DIMENSION: TEMPERATURE INDEX ('TOUT')
C                            2nd DIMENSION: TRANSITION INDEX
C          (R*8)  EXCRP(,) = PROTON IMPACT TRANSITION:
C                            EXCITATION RATE COEFFS (cm**3/s)
C                            PRE  'BXRATE': UNIT GAMMA VALUES
C                            POST 'BXRATE': TRUE VALUES
C                            1st DIMENSION: TEMPERATURE INDEX ('TOUT')
C                            2nd DIMENSION: TRANSITION INDEX
C          (R*8)  DEXCRP(,)= PROTON IMPACT TRANSITION:
C                            DE-EXCITATION RATE COEFFS (cm**3/s)
C                            PRE  'BXRATE': UNIT GAMMA VALUES
C                            POST 'BXRATE': TRUE VALUES
C                            1st DIMENSION: TEMPERATURE INDEX ('TOUT')
C                            2nd DIMENSION: TRANSITION INDEX
C          (R*8)  VECH(,)  = CHARGE-EXCHANGE RECOMBINATION:
C                            SPLINED RECOMBINATION RATE COEFFT. VALUES.
C                            1st DIMENSION: TEMPERATURE INDEX ('TOUT')
C                            2nd DIMENSION: CAPTURING LEVEL INDEX.
C          (R*8)  VECR(,)  = FREE ELECTRON RECOMBINATION:
C                            SPLINED RECOMBINATION RATE COEFFT. VALUES.
C                            1st DIMENSION: TEMPERATURE INDEX ('TOUT')
C                            2nd DIMENSION: CAPTURING LEVEL INDEX.
```

```
C          (R*8)  CRA(,)    = A-VALUE  (sec-1)  MATRIX  COVERING  ALL
C                              TRANSITIONS.
C                              1st DIMENSION: ENERGY LEVEL INDEX
C                              2nd DIMENSION: ENERGY LEVEL INDEX
C                             (NOTE: DIAGONAL   ELEMENTS  REPRESENT   THE
C                                    NEGATIVE  SUM  OF  THEIR  RESPECTIVE
C                                    COLUMNS.)
C          (R*8)  CRCE(,)   = ELECTRON IMPACT TRANSITION:
C                              EXCIT'N/DE-EXCIT'N RATE COEFFT MATRIX
C                              COVERING ALL TRANSITIONS (cm**3/s).
C                              VALUES FOR GIVEN TEMPERATURE & TRANSITION
C                              TYPE.
C                              1st DIMENSION: ENERGY LEVEL INDEX
C                              2nd DIMENSION: ENERGY LEVEL INDEX
C                             (NOTE: DIAGONAL   ELEMENTS  REPRESENT   THE
C                                    NEGATIVE  SUM  OF  THEIR  RESPECTIVE
C                                    COLUMNS.)
C          (R*8)  CRCP(,)   = PROTON IMPACT TRANSITION:
C                              EXCIT'N/DE-EXCIT'N RATE COEFFT MATRIX
C                              COVERING ALL TRANSITIONS (cm**3/s).
C                              VALUES FOR GIVEN TEMPERATURE & TRANSITION
C                              TYPE.
C                              1st DIMENSION: ENERGY LEVEL INDEX
C                              2nd DIMENSION: ENERGY LEVEL INDEX
C                             (NOTE: DIAGONAL   ELEMENTS  REPRESENT   THE
C                                    NEGATIVE  SUM  OF  THEIR  RESPECTIVE
C                                    COLUMNS.)
C          (R*8)  CC(,)     = RATE MATRIX COVERING ALL TRANSITIONS
C                              (UNITS: SEC-1)
C                              VALUES FOR GIVEN TEMPERATURE AND DENSITY.
C                              1st DIMENSION: ENERGY LEVEL INDEX
C                              2nd DIMENSION: ENERGY LEVEL INDEX
C          (R*8)  CMAT(,)   = (INVERTED)   RATE  MATRIX  COVERING   ALL
C                              NON-METASTABLE/ORDINARY EXCITED LEVELS.
C                              (UNITS: SEC)
C                              VALUES FOR GIVEN TEMPERATURE AND DENSITY.
C                              PRE  'XXMINV' : NOT-INVERTED
C                              POST 'XXMINV' : INVERTED
C                              1st DIMENSION: ORDINARY EXCITED LEVEL INDEX
C                              2nd DIMENSION: ORDINARY EXCITED LEVEL INDEX
C          (R*8)  CRED(,)   = MATRIX  OF  TRANSITION   RATES   BETWEEN
C                              METASTABLE LEVELS.
C                              (UNITS: SEC-1)
C                              VALUES FOR GIVEN TEMPERATURE AND DENSITY.
C                              1st DIMENSION: METASTABLE LEVEL INDEX
C                              2nd DIMENSION: METASTABLE LEVEL INDEX
C          (R*8)  CRMAT(,) = INVERTED  METASTABLE  LEVEL   RATE   MATRIX
C                              COVERING ALL TRANSITIONS BETWEEN METASTABLE
C                              LEVELS EXCEPT THOSE INVOLVING LEVEL 1.
C                              VALUES FOR GIVEN TEMPERATURE AND DENSITY.
C                              BEFORE INPUT  TO   XXMINV: NOT INVERTED
C                              AFTER  OUTPUT FROM XXMINV: AS-ABOVE
C                              1st DIMENSION: METASTABLE LEVEL INDEX - 1
C                              2nd DIMENSION: METASTABLE LEVEL INDEX - 1
C
C
C          (R*8) POPAR(,)   = LEVEL POPULATIONS
C                               1st DIMENSION: LEVEL INDEX
C                               2nd DIMENSION: TEMPERATURE INDEX
C          (R*8) STVR(,)    = ORDINARY EXCITED LEVEL:
C                              FREE-ELECTRON RECOMBINATION COEFFICIENTS
```

```
C                              (UNITS* CM**3/SEC-1)
C                              1st DIMENSION: ORDINARY LEVEL INDEX
C                              2nd DIMENSION: TEMPERATURE INDEX
C          (R*8) STVH(,)    = ORDINARY EXCITED LEVEL:
C                              CHARGE-EXCHANGE RECOMBINATION COEFFICIENTS
C                              (UNITS* CM**3/SEC-1)
C                              1st DIMENSION: ORDINARY LEVEL INDEX
C                              2nd DIMENSION: TEMPERATURE INDEX
C          (R*8) STACK(,,) = ORDINARY EXCITED LEVEL POPULAT'N DEPENDENCE
C                            ON METASTABLE LEVEL.
C                              1st DIMENSION: ORDINARY LEVEL INDEX
C                              2nd DIMENSION: METASTABLE INDEX
C                              3rd DIMENSION: TEMPERATURE INDEX
C          (R*8) STVRM(,)   = METASTABLE LEVEL:
C                              FREE-ELECTRON RECOMBINATION COEFFICIENTS
C                              (UNITS* CM**3/SEC-1)
C                              1st DIMENSION: METASTABLE INDEX
C                              2nd DIMENSION: TEMPERATURE INDEX
C          (R*8) STVHM(,)   = METASTABLE LEVEL:
C                              CHARGE-EXCHANGE RECOMBINATION COEFFICIENTS
C                              (UNITS* CM**3/SEC-1)
C                              1st DIMENSION: METASTABLE INDEX
C                              2nd DIMENSION: TEMPERATURE INDEX
C          (R*8) STCKM(,)   = METASTABLE POPULATIONS STACK
C                              1st DIMENSION: METASTABLE INDEX
C                              2nd DIMENSION: TEMPERATURE INDEX
C
C          (C*1)  TCODE() = TRANSITION: DATA TYPE POINTER:
C                           ' ' => Electron Impact   Transition
C                           'P' => Proton   Impact   Transition
C                           'H' => Charge   Exchange Recombination
C                           'R' => Free     Electron Recombination
C          (C*18) CSTRGA()= NOMENCLATURE/CONFIGURATION FOR LEVEL 'IA()'
C          (C*22) STRGA() = LEVEL DESIGNATIONS
C
C          (L*4)  LTRNG(,)= .TRUE.  => TEMPERATURE VALUE WITHIN RANGE
C                                      READ FROM INPUT COPASE DATA SET.
C                         = .FALSE. =>TEMPERATURE VALUE NOT WITHIN RANGE
C                                      READ FROM INPUT COPASE DATA SET.
C                           1st DIMENSION: TEMPERATURE INDEX.
C                           2nd DIMENSION: TEMPERATURE TYPE -
C                                          1) => ELECTRON
C                                          2) => PROTON
C                                          3) => NEUTRAL HYDROGEN
C
C NOTE:
C
C          INPUT/OUTPUT STREAM ALLOCATIONS:
C          --------------------------------
C
C          STREAM 10: INPUT - SPECIFIC ION RATE DATA INPUT FILE FROM
C          ('IUNT10')         DATABASE (SEE DATA SECTION ABOVE).
C
C
C AUTHOR:  HP SUMMERS
C          K1/1/57
C          JET EXT. 4941
C
C DATE:    27/06/91
C
C UPDATE:  12/04/94 - H. P. SUMMERS - RATIONALISING OF DIMENSIONS WITH
```

```
C                                     LATEST ADAS9120 ROUTINES. NOTED
C                                     STACK IS REAL*4
C
C#
C DATE:    mar20-95 - A. C. Lanzafame - conversion to Unix
C          mar21-95 - A. C. Lanzafame - call to XXDATE avoided: redundant
C          mar24-95 -                 - FILELS from C*44 to C*60
C                                     - DSNINC from C*44 to C*60
C          apr27-95 - A. C. Lanzafame - STACK changed to R*8
C
C  ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C      DCSTKC      ADAS      STACK UP TRANSITION RATE BETWEEN METS.
C      DCSTKA      ADAS      STACK UP ORDINARY POP. DEPENDENCE ON MET
C      DCPOPM      ADAS      CALCULATE BASIC MET. LEVEL POPULATIONS.
C      DCPOPO      ADAS      CALCULATE ORDINARY LEVEL POPULATIONS.
C      DCLNORM     ADAS      NORMALISES LINE EMISSIVITY.
C      BXDATA      ADAS      GATHERS RELEVANT DATA FROM INPUT FILE
C      BXTTYP      ADAS      SORT TRANSITIONS INTO TRAN/RECOMB TYPES
C      BXIORD      ADAS      SETS  UP ORDINARY LEVEL INDEX.
C      BXRATE      ADAS      CALCULATES EXC. & DE-EXC. RATE COEFFTS.
C      BXRCOM      ADAS      ESTABLISHES RECOMBINATION RATE COEFFTS.
C      BXMCRA      ADAS      CONSTRUCTS A-VALUE MATRIX.
C      BXMCRC      ADAS      CONSTRUCTS EXC./DE-EXC. RATE COEF MATRIX
C      BXMCCA      ADAS      CONSTRUCTS WHOLE RATE MATRIX.
C      BXMCMA      ADAS      CONSTRUCTS ORDINARY LEVEL RATE MATRIX.
C      BXSTKB      ADAS      STACK UP RECOMB. CONTRIBUTION FOR ORD.
C      BXSTKD      ADAS      STACK UP RECOMB RATE FOR EACH MET. LEVEL
C      BXMPOP      ADAS      CALCULATE METASTABLE LEVEL POPULATIONS.
C      BXSTVM      ADAS      CALCULATE MET. LEVEL RECOMB. COEFFTS.
C      XXERYD      ADAS      CONVERTS ENERGIES FROM W.NO. TO RYDBERGS
C      XXRATE      ADAS      CALCULATES EXC. & DE-EXC. RATE COEFFTS.
C      XXMINV      ADAS      INVERTS MATRIX AND SOLVES EQUATIONS.
C                                FOR UNIT GAMMA VALUE
C      R8FBCH      ADAS      REAL*8 FUNCTION:EVALUATES SHELL CONTRIB.
C                                TO IONISATION RATE  COEFFICIENT  IN  THE
C                                BURGESS-CHIDICHIMO APPROX.
C
C VERSION 1.1 DATE: 18-06-98
C STUART LOCH
C - BASED ON DCXCOEF.FOR (adas412)
C VERSION 1.2 DATE: 24-09-99
C STUART LOCH
C - NDTRN INCREASED FROM 1100 TO 2000 TO ALLOW FOR
C                LONGER ADF04 FILES TO BE PROCESSED.
C
C VERSION : 1.3
C DATE    : 02-05-2003
C MODIFIED: Martin O'Mullane
C - Use xxdata_04 to read adf04 file. This requires
C                new arrays some of which are not used in the
C                population calculation.
C              - bxttyp parameter list extended.
C
C VERSION : 1.4
C DATE    : 15-03-2005
C MODIFIED: Martin O'Mullane
C - Increase ndmet to 4 in order to be able to read
C                adf04 datasets from GCR Project.
C - files() (and dsninc) increased to character*80.
```

```
C
C-----------------------------------------------------------------------
      CHARACTER*80        FILELS(NZDIM)
      CHARACTER*51        SPEC(NDLEV,NZDIM)
      INTEGER             IFAIL,       INDA(NDTRN,NZDIM),        MAXT
      INTEGER             NIND(NZDIM), NION,        NSPEC(NZDIM)
      LOGICAL             LFXIST(NZDIM),            LHSEL,       LISEL
      LOGICAL             LPSEL,       LRSEL,       LZSEL
      REAL*8              COEF(NDTRN,NDTEM,NZDIM),  DENSA(NDTEM)
      REAL*8              DENSPA(NDTEM),            POPAR(NDLEV,NDTEM)
      REAL*8              RATHA(NDTEM),             RATIA(NDTEM)
      REAL*8              TEVA(NDTEM), THVA(NDTEM), TPVA(NDTEM), ZEFF
```

3.82 bfdata: Subroutine bfdata from library adas2xx

```
C
       SUBROUTINE BFDATA( IUNIT  , NDLEV  , NDTRN , NDMET ,
     &                    TITLED , IZ     , IZ0   , IZ1   , BWNO  ,
     &                    NPL    , BWNOA  , LBSETA, PRTWTA, CPRTA ,
     &                    IL     ,
     &                    IA     , CSTRGA , ISA   , ILA   , XJA   , WA ,
     &                    CPLA   , NPLA   , IPLA  , ZPLA  ,
     &                    CIONP  ,
     &                    NV     , SCEF   ,
     &                    ITRAN  , MAXLEV ,
     &                    TCODE  , I1A    , I2A   , AVAL  , SCOM
     &                  )
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: BFDATA *********************
C
C  PURPOSE:  TO FETCH DATA FROM INPUT ADF04 DATA SET, INCLUDING
C            MULTIPLE PARENTS ON FREE-ELECTRON AND CHARGE EXCHANGE
C            ON RECOMBINATION, INCLUSION OF EXPLICIT CONTRIBUTIONS BY
C            IONISATION.
C
C            IMPROVEMENT OF AUTOMATIC IONISATION CALC. BY INCLUDING
C            ASSIGNMENT OF FINAL STATE PARENT. KEEPS IONISATION
C            POTENTIAL INFORMATION FROM LEVEL TERMINATOR LINE.
C
C  CALLING PROGRAM: ADAS215
C
C  DATA:
C           THE 'REAL' DATA IN THE FILE IS REPRESENTED IN AN ABBREVIATED
C           FORM WHICH OMITS THE "D" OR "E" EXPONENT SPECIFIER.
C           e.g. 1.23D-06 or 1.23E-06 IS REPRESENTED AS 1.23-06
C                6.75D+07 or 6.75E+07 IS REPRESENTED AS 6.75+07
C
C           THEREFORE THE FORM OF EACH 'REAL' NUMBER IN THE DATA SET IS:
C                          N.NN+NN or N.NN-NN
C
C           THE UNITS USED IN THE DATA FILE ARE TAKEN AS FOLLOWS:
C
C           IONISATION POTENTIAL: WAVE NUMBER (CM-1)
C           INDEX LEVEL ENERGIES: WAVE NUMBER (CM-1)
C           TEMPERATURES        : KELVIN
C           A-VALUES            : SEC-1
C           GAMMA-VALUES        :
C           RATE COEFFT.        : CM3 SEC-1
C
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  IUNIT   = UNIT TO WHICH INPUT FILE IS ALLOCATED
C  INPUT : (I*4)  NDLEV   = MAXIMUM NUMBER OF LEVELS THAT CAN BE READ
C  INPUT : (I*4)  NDTRN   = MAX. NUMBER OF TRANSITIONS THAT CAN BE READ
C  INPUT : (I*4)  NDMET   = MAX. NUMBER OF METASTABLES ALLOWED
C
C  OUTPUT: (C*3)  TITLED  = ELEMENT SYMBOL.
C  OUTPUT: (I*4)  IZ      =  RECOMBINED ION CHARGE READ
C  OUTPUT: (I*4)  IZ0     =          NUCLEAR CHARGE READ
C  OUTPUT: (I*4)  IZ1     = RECOMBINING ION CHARGE READ
C                           (NOTE: IZ1 SHOULD EQUAL IZ+1)
```

```
C  OUTPUT: (R*8)  BWNO    = IONISATION POTENTIAL (CM-1) OF LOWEST PARENT
C  OUTPUT: (I*4)  NPL     = NUMBER OF PARENTS ON FIRST LINE AND USED
C                           IN LEVEL ASSIGNMENTS
C  OUTPUT: (R*8)  BWNOA() = IONISATION POTENTIAL (CM-1) OF PARENTS
C  OUTPUT: (L*4)  LBSETA()= .TRUE.  - PARENT WEIGHT SET FOR BWNOA()
C                           .FALSE. - PARENT WEIGHT NOT SET FOR BWNOA()
C  OUTPUT: (R*8)  PRTWTA()= PARENT WEIGHT FOR BWNOA()
C  OUTPUT: (C*9)  CPRTA() = PARENT NAME IN BRACKETS
C
C  OUTPUT: (I*4)  IL      = INPUT DATA FILE: NUMBER OF ENERGY LEVELS
C
C  OUTPUT: (I*4)  IA()    = ENERGY LEVEL INDEX NUMBER
C  OUTPUT: (C*18) CSTRGA()= NOMENCLATURE/CONFIGURATION FOR LEVEL 'IA()'
C  OUTPUT: (I*4)  ISA()   = MULTIPLICITY FOR LEVEL 'IA()'
C                           NOTE: (ISA-1)/2 = QUANTUM NUMBER (S)
C  OUTPUT: (I*4)  ILA()   = QUANTUM NUMBER (L) FOR LEVEL 'IA()'
C  OUTPUT: (R*8)  XJA()   = QUANTUM NUMBER (J-VALUE) FOR LEVEL 'IA()'
C                           NOTE: (2*XJA)+1 = STATISTICAL WEIGHT
C  OUTPUT: (R*8)  WA()    = ENERGY RELATIVE TO LEVEL 1 (CM-1) FOR LEVEL
C                           'IA()'
C  OUTPUT: (C*1)  CPLA()  = CHAR. SPECIFYING 1ST PARENT FOR LEVEL 'IA()'
C                                INTEGER - PARENT IN BWNOA() LIST
C                                'BLANK' - PARENT BWNOA(1)
C                                  'X'   - DO NOT ASSIGN A PARENT
C  OUTPUT: (I*4)  NPLA()  = NO. OF PARENT/ZETA CONTRIBUTIONS TO IONIS.
C                           OF LEVEL
C  OUTPUT: (I*4)  IPLA(,) = PARENT INDEX FOR CONTRIBUTIONS TO IONIS.
C                           OF LEVEL
C                           1ST DIMENSION: PARENT INDEX
C                           2ND DIMENSION: LEVEL INDEX
C  OUTPUT: (I*4)  ZPLA(,) = EFF. ZETA PARAM. FOR CONTRIBUTIONS TO IONIS.
C                           OF LEVEL
C                           1ST DIMENSION: PARENT INDEX
C                           2ND DIMENSION: LEVEL INDEX
C
C  OUTPUT: (C*92) CIONP   = STRING CONTAINING LEVEL TERMINATOR LINE
C
C  OUTPUT: (I*4)  NV      = INPUT DATA FILE: NUMBER OF GAMMA/TEMPERATURE
C                           PAIRS FOR A GIVEN TRANSITION.
C  OUTPUT: (R*8)  SCEF()  = INPUT DATA FILE: ELECTRON TEMPERATURES (K)
C                           (INITIALLY JUST THE MANTISSA. SEE 'ITPOW()')
C                           (NOTE: TE=TP=TH IS ASSUMED)
C
C  OUTPUT: (I*4)  ITRAN   = INPUT DATA FILE: NUMBER OF TRANSITIONS
C  OUTPUT: (I*4)  MAXLEV  = HIGHEST INDEX LEVEL IN READ TRANSITIONS
C
C  OUTPUT: (C*1)  TCODE() = TRANSITION: DATA TYPE POINTER:
C                           ' ' => Electron Impact   Transition
C                           'P' => Proton   Impact   Transition
C                           'H' => Charge   Exchange Recombination
C                           'R' => Free     Electron Recombination
C                           'I' => Coll. ionisation from lower stage ion
C  OUTPUT: (I*4)  I1A()   = TRANSITION:
C                            LOWER ENERGY LEVEL INDEX (CASE ' ' & 'P')
C                            SIGNED PARENT NDEX (CASE 'H','R' & 'I')
C  OUTPUT: (I*4)  I2A()   = TRANSITION:
C                            UPPER ENERGY LEVEL INDEX (CASE ' ' & 'P')
C                            CAPTURING LEVEL INDEX (CASE 'H','R' & 'I')
C  OUTPUT: (R*8)  AVAL()  = TRANSITION:
C                            A-VALUE (SEC-1)          (CASE ' ')
C                            NEUTRAL BEAM ENERGY      (CASE 'H')
```

```
C                              NOT USED             (CASE 'P','R' & 'I')
C  OUTPUT: (R*8)  SCOM(,) = TRANSITION:
C                              GAMMA VALUES             (CASE ' ' & 'P')
C                              RATE COEFFT.(CM3 SEC-1)(CASE 'H','R' & 'I')
C                             1ST DIMENSION - TEMPERATURE 'SCEF()'
C                             2ND DIMENSION - TRANSITION NUMBER
C
C          (I*4)  NVMAX   = PARAMETER = MAX. NUMBER OF TEMPERATURES
C                                       THAT CAN BE READ IN.
C          (I*4)  MTIED   = PARAMETER = MUST BE GREATER THAN OR EQUAL TO
C                                       THE MAX. NO. OF LEVELS.
C          (R*8)  DZERO   = PARAMETER = MINIMUM VALUE FOR 'AVAL()' AND
C                                       'SCOM()' ARRAYS = 1.0D-30
C
C          (I*4)  I4UNIT  = FUNCTION (SEE ROUTINE SELECTION BELOW)
C          (I*4)  IQS     = X-SECT DATA FORMAT SELECTOR
C                           NOTE: IQS=3 ONLY ALLOWED IN THIS PROGRAM
C          (I*4)  IFAIL   = FAILURE NUMBER FROM B9PARS AND B9PRS1
C          (I*4)  I       = GENERAL USE.
C          (I*4)  IABT    = RETURN CODE FROM 'R(FCTN' (0 => NO ERROR)
C                           OR FROM INTERROGATION OF 'C7'
C          (I*4)  J       = GENERAL USE.
C          (I*4)  J1      = INPUT DATA FILE - SELECTED TRANSITION:
C                            LOWER ENERGY LEVEL INDEX (CASE ' ' & 'P')
C          (I*4)  J2      = INPUT DATA FILE - SELECTED TRANSITION:
C                            UPPER ENERGY LEVEL INDEX (CASE ' ' & 'P')
C                            CAPTURING    LEVEL INDEX (CASE 'H' & 'R')
C          (I*4)  LENCST  = BYTE LENGTH OF STRING CSTRGA()
C          (I*4)  ILINE   = ENERGY LEVEL INDEX FOR CURRENT LINE
C          (I*4)  IRECL   = RECORD LENGTH OF INPUT DATASET (<=128)
C          (I*4)  IAPOW   = EXPONENT OF 'AVALM'
C          (I*4)  IGPOW() = EXPONENT OF 'GAMMA()'
C          (I*4)  ITPOW() = TEMPERATURES - EXPONENT
C                           NOTE: MANTISSA INITIALLY KEPT IN 'SCEF()'
C
C          (R*4)  ZF      = SHOULD BE EQUIVALENT TO 'IZ1'
C
C          (R*8)  AVALM   = INPUT DATA FILE - SELECTED TRANSITION:
C                           MANTISSA OF:   ('IAPOW' => EXPONENT)
C                            A-VALUE (SEC-1)          (CASE ' ')
C                            NEUTRAL BEAM ENERGY      (CASE 'H')
C                            NOT USED              (CASE 'P','R' & 'I')
C          (R*8)  GAMMA() = INPUT DATA FILE - SELECTED TRANSITION:
C                           MANTISSA OF: ('IGPOW()' => EXPONENT)
C                            GAMMA VALUES             (CASE ' ' & 'P')
C                            RATE COEFFT.(CM3 SEC-1)(CASE 'H','R' & 'I')
C                           DIMENSION => TEMPERATURE 'SCEF()'
C
C          (C*7)  C7      = USED TO PARSE VALUE FOR XJA()
C          (C*7)  CDELIM  = DELIMITERS FOR INPUT OF DATA FROM HEADERS
C          (C*18) C18     = USED TO PARSE VALUE TO CSTRGA()
C          (C*18) C18T    = COPY OF C18 : UNSATISFACTORY METHOD OF
C                           AVOIDING COMPILER REFERENCE ERROR :
C                           DHB 07.04.95
C          (C*80) CLINE   = CURRENT ENERGY LEVEL INDEX PARAMETER LINE
C          (C*75) STRING  = TAIL STRING OF 1ST DATA LINE FOR PARSING
C          (C*44) STRG1   = TAIL STRING OF LEVEL SPEC LINES FOR PARSING
C          (C*128)BUFFER  = GENERAL STRING BUFFER STORAGE
C          (C*3)  CITPOW()= USED TO PARSE VALUES TO ITPOW()
C          (C*5)  CSCEF() = USED TO PARSE VALUES TO SCEF()
C
```

```
C          (L*4)  LDATA   = IDENTIFIES  WHETHER  THE END OF AN  INPUT
C                           SECTION IN THE DATA SET HAS BEEN LOCATED.
C                           (.TRUE. => END OF SECTION REACHED)
C          (L*4)  LTCHR   = .TRUE.  => CURRENT 'TCODE()' = 'H' OR 'R'
C                                                          OR 'I'
C                         = .FALSE. => CURRENT 'TCODE()'.NE.'H' OR 'R'
C                                                           OR 'I'
C          (L*4)  LTCPR   = .TRUE.  => CURRENT 'TCODE()' = 'P' OR 'R'
C                                                          OR 'I'
C                         = .FALSE. => CURRENT 'TCODE()'.NE. 'P' OR 'R'
C                                                            OR 'I'
C          (L*4)  LERROR  = .TRUE.  => UNTIED LEVEL FOUND
C                         = .FALSE. => ALL LEVELS TIED
C          (L*4)  LTIED() = .TRUE.  => SPECIFIED LEVEL TIED
C                         = .FALSE. => SPECIFIED LEVEL IS UNTIED
C                           DIMENSION => LEVEL INDEX
C
C
C NOTE:             LTCHR        LTCPR         TCODE()
C                  ----------------------------------
C                  .TRUE.       .TRUE.    =>     'R','I'
C                  .TRUE.       .FALSE.   =>     'H'
C                  .FALSE.      .TRUE.    =>     'P'
C                  .FALSE.      .FALSE.   =>     ' '
C
C        FOR A-VALUES & GAMMA-VALUES ENTRIES LESS THAN 'DZERO' ARE TAKEN
C        AS BEING EQUAL TO DZERO. THIS AFFECTS THE 'AVAL()' AND 'SCOM()'
C        ARRAYS.
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4UNIT     ADAS      FETCH UNIT NUMBER FOR OUTPUT OF MESSAGES
C          R8FCTN     ADAS      CONVERTS FROM CHARACTER TO REAL VARIABLE
C          I4FCTN     ADAS      CONVERTS FROM CHAR. TO INTEGER  VARIABLE
C          XXWORD     ADAS      PARSES A STRING INTO SEPARATE WORDS
C                               FOR ' ()<>{}' DELIMITERS
C
C AUTHOR:  HP SUMMERS   UNIVERSITY OF STRATHCLYDE
C          JA8.08
C          TEL.  0141-553-4196
C
C DATE:    04/06/98
C
C UPDATE:  1.1 DATE: 09/08/98
C MODIFIED: RICHARD MARTIN
C - PUT UNDER SCCS CONTROL.
C
C UPDATE:  1.2 DATE: 17/05/07
C MODIFIED: Allan Whiteford
C - Updated comments as part of subroutine documentation
C                 procedure.
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      CHARACTER*92        CIONP
      CHARACTER           CPLA(NDLEV)
      CHARACTER*9         CPRTA(NDMET)
      CHARACTER*(*)       CSTRGA(NDLEV)
      CHARACTER           TCODE(NDTRN)
      CHARACTER*3         TITLED
```

```
INTEGER            I1A(NDTRN),  I2A(NDTRN),  IA(NDLEV),    IL
INTEGER            ILA(NDLEV),  IPLA(NDMET,NDLEV)
INTEGER            ISA(NDLEV),  ITRAN,       IUNIT,        IZ
INTEGER            IZ0,         IZ1,         MAXLEV,       NDLEV
INTEGER            NDMET,       NDTRN,       NPL
INTEGER            NPLA(NDLEV), NV
LOGICAL            LBSETA(NDMET)
REAL*8             AVAL(NDTRN), BWNO,        BWNOA(NDMET)
REAL*8             PRTWTA(NDMET),            SCEF(NVMAX)
REAL*8             SCOM(NVMAX,NDTRN),        WA(NDLEV)
REAL*8             XJA(NDLEV),  ZPLA(NDMET,NDLEV)
```

3.83 bftran: Subroutine bftran from library adas2xx

```
C
      SUBROUTINE  BFTRAN( TYP   , C      , LNEG  ,
     &                    AIN   , WVNOU  , WVNOL , WTU   , WTL  ,
     &                    TEIN  , UPSIN  , NV    ,
     &                    TEOUT , UPSOUT , NVN
     &                  )
C-----------------------------------------------------------------
C
C ************* FORTRAN SUBROUTINE  BFTRAN  ******************
C
C PURPOSE : TO IMPLEMENT THE TRANSFORMATION DESCRIBED BY
C           BURGESS AND TULLY ( SEE REFERENCE (1)) WHICH
C           IS USED TO ASSESS AND COMPACT DATA.
C
C
C REFERENCES:
C           (1) A.BURGESS AND J.A.TULLY
C           ON THE ANALYSIS OF COLLISION STRENGTHS
C           AND RATE COEFFICIENTS.
C           ASTRON.ASTROPHYS.254,436-453 (1992 )
C
C           (2) SUMMERS.H.P
C               ADAS USERS MANUAL ( 1ST EDITION ).
C
C INPUT :
C       (C*1) TYP         = BURGESS & TULLY TRANSITION TYPE CODE
C       (R*8) C           = THE ADJUSTABLE PARAMETER ASSOCIATED
C                           WITH THE BURGESS AND TULLY
C                           TRANSFORMATION ( SEE REFERENCE (1)).
C       (L)   LNEG        = IF TYPE 2 GOES NEGATIVE AT HIGH
C                           TEMPERATURES THE COLLISION STRENGTH HAS
C                           A ZERO LIMITING VALUE AT INFINITY.
C       (R*8) AIN         = THE EINSTEIN 'A' CO-EFFICIENT. THIS
C                           IS READ DIRECTLY FROM THE ADF04
C                           TYPE FILE.
C       (R*8) WVNOU       = THE WAVENUMBER OF THE UPPER LEVEL.
C                           THIS IS READ DIRECTLY FROM THE ADF04
C                           TYPE FILE.
C       (R*8) WVNLO       = THE WAVENUMBER OF THE LOWER LEVEL.
C                           THIS IS READ DIRECTLY FROM THE ADF04
C                           TYPE FILE.
C       (R*8) WTU         = THE STATISTICAL WEIGHT OF THE UPPER
C                           LEVEL. THIS IS OBTAINED BY
C
C       (R*8) WTL         = THE STATISTICAL WEIGHT OF THE LOWER
C                           LEVEL. THIS IS OBTAINED BY
C
C       (R*8) TEIN        = THE TEMPERATURE ARRAY (K). THIS
C                           DATA IS READ DIRECTLY FROM THE
C                           ADF04 TYPE FILE.
C       (R*8) UPSIN       = THE ARRAY CONTAING THE EFFECTIVE
C                           COLLISION STRENGTH. THIS DATA IS
C                           READ DIRECTLY FROM THE ADF04
C                           TYPE FILE.
C       (I*4) NV          = THE NUMBER OF TEMPERATURE/EFFECTIVE
C                           COLLISION STRENGTH PAIRS FOR A GIVEN
C                           TRANSITION.
C       (I*4) NVN         = THE NUMBER OF TEMPERATURES/EFFECTIVE
C                           COLLISION STRENGTH PAIRS FOR A GIVEN
```

```
C                           TRANSITION. THIS PARAMETER IS IN
C                           FACT THE NUMBER OF USER DEFINED
C                           TEMPERATURE POINTS AT WHICH THE
C                           EFFECTIVE COLLISION STRENGTH
C                           HAS TO BE EVALUATED AT.
C       (R*8) TEOUT       = THE TEMPERATURE RANGE FOR WHICH
C                           THE EFFECTIVE COLLISION STRENGTH IS
C                           REQUIRED.
C
C OUTPUT:
C
C       (R*8) UPSOUT      = THE ARRAY OF EFFECTIVE COLLISION
C                           STRENGTHS THAT ARE REQUIRED.
C
C
C       (R*8) E           = THE MATHEMATICAL CONSTANT E.
C       (R*8) CONST       = CLUSTER OF PHYSICAL CONSTANTS.
C                           SEE PAGE 12 OF REFERENCE (2).
C       (R*8) EIJIN       = THE TRANSITION ENERGY (RYD).
C       (R*8) FIJIN       = THE OSCILLATOR STRENGTH.
C       (R*8) ET          = GENERAL CONSTANT.
C       (R*8) C           = THE BURGESS C PARAMETER.
C       (R*8) X           = THE X ARRAY ASSOCIATED WITH THE
C                           BURGESS AND TULLY TRANSFORMATION.
C       (R*8) Y           = THE Y ARRAY ASSOCIATED WITH THE
C                           BURGESS AND TULLY TRANSFORMATION.
C       (R*8) DY          = DERIVATIVES AT INPUT KNOTS.
C                           SEE XXSPLN FOR FUTHER DETAILS.
C       (R*8) XOUT        = X ARRAY ASSOCIATED WITH THE
C                           BURGESS AND TULLY TRANSFORMATION.
C       (R*8) YOUT        = Y ARRAY ASSOCIATED WITH THE
C                           BURGESS AND TULLY TRANSFORMATION.
C       (R*8) FINTX       = INTERPOLATING X COORDINATE
C                           TRANSFORMATION ( SEE SUBROUTINE
C                           XXSPLN ). EXTERNAL FUNCTION.
C       (I*4) NVMAX       = THE MAXIMUM NUMBER OF TEMPERTURES
C                           THAT CAN BE READ.
C       (I*4) NFIT        = NVMAX+1 - ALLOWS LIMIT POINT TO BE
C                           ADDED TO TYPE 1 AND 4 FITS.
C       (I*4) IOPT        = GENERAL PARAMETER ASSOCIATED WITH
C                           THE SUBROUTINE XXSPLN.
C       (I*4) I           = GENERAL VARIABLE WHICH IS USED AS
C                           A COUNTER.
C       (LOG) LSETX       = PARAMETER ASSOCIATED WITH THE
C                           SUBROUTINE XXSPLN.
C
C  ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4UNIT     ADAS      FETCH UNIT NUMBER FOR OUTPUT OF MESSAGES
C          R8CONST    ADAS      RETURNS FUNDAMENTAL ATOMIC CONSTANTS
C
C
C AUTHOR:   H. P. SUMMERS, UNIVERSITY OF STRATHCLYDE
C           JA8.08
C           TEL.  0141-553-4196
C
C DATE:    04/06/98
C
C UPDATE:
C
```

```
C VERSION:  1.1 DATE: 09/08/98
C MODIFIED: RICHARD MARTIN
C - PUT UNDER SCCS CONTROL.
C
C VERSION:  1.2 DATE: 15/04/99
C MODIFIED: Martin O'Mullane
C               - If type 1 or 4 add in the limit point
C                 to the spline fit.
C - Added support for type 4 transitions.
C
C VERSION:  1.3 DATE: 08/10/99
C MODIFIED: Martin O'Mullane
C               - Certain type 2 and 3 transitions have a limiting
C                 point of zero at infinity. We deduce this by
C                 fitting and if the values go negative rerun with
C                 LNEG set and add y=0 at x=1 to the fit.
C
C-----------------------------------------------------------------
      CHARACTER           TYP
      INTEGER             NV,          NVN
      LOGICAL             LNEG
      REAL*8              AIN,         C,           TEIN(NV)
      REAL*8              TEOUT(NVN),  UPSIN(NV),   UPSOUT(NVN), WTL
      REAL*8              WTU,         WVNOL,       WVNOU
```

3.84 bfttyp: Subroutine bfttyp from library adas2xx

```
C
      SUBROUTINE BFTTYP( NDLEV  , NDTRN  ,
     &                   IZ1    , IL     ,
     &                   IA     , CSTRGA , ISA    , ILA   , XJA   , WA ,
     &                   ITRAN  , TCODE  , I1A    , I2A   , AVAL  ,
     &                   ICNTE  , ICNTP  , ICNTR  , ICNTH , ICNTI ,
     &                   IETRN  , PECODE , TECODE , IE1A  , IE2A  , AA ,
     &                   CEA
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: BFTTYP *********************
C
C  PURPOSE:  TO SORT TRANSITION ARRAYS INTO FOUR TRANSITION/RECOMB TYPES
C            AND ASSIGN INITIAL TYPES TO ELECTRON IMPACT TRANSITIONS
C
C  CALLING PROGRAM: ADAS215
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  NDLEV   = MAXIMUM NUMBER OF LEVELS THAT CAN BE READ
C  INPUT : (I*4)  NDTRN   = MAX. NUMBER OF TRANSITIONS THAT CAN BE READ
C
C  INPUT : (I*4)  IZ1     = RECOMBINING ION CHARGE READ
C  INPUT : (I*4)  IL      = INPUT DATA FILE: NUMBER OF ENERGY LEVELS
C
C  INPUT : (I*4)  IA()    = ENERGY LEVEL INDEX NUMBER
C  INPUT : (C*18) CSTRGA()= NOMENCLATURE/CONFIGURATION FOR LEVEL 'IA()'
C  INPUT : (I*4)  ISA()   = MULTIPLICITY FOR LEVEL 'IA()'
C                           NOTE: (ISA-1)/2 = QUANTUM NUMBER (S)
C  INPUT : (I*4)  ILA()   = QUANTUM NUMBER (L) FOR LEVEL 'IA()'
C  INPUT : (R*8)  XJA()   = QUANTUM NUMBER (J-VALUE) FOR LEVEL 'IA()'
C                           NOTE: (2*XJA)+1 = STATISTICAL WEIGHT
C  INPUT : (R*8)  WA()    = ENERGY RELATIVE TO LEVEL 1 (CM-1) FOR LEVEL
C                           'IA()'
C
C  INPUT : (I*4)  ITRAN   = INPUT DATA FILE: NUMBER OF TRANSITIONS
C  INPUT : (C*1)  TCODE() = TRANSITION: DATA TYPE POINTER:
C                           ' ' => Electron Impact   Transition
C                           'P' => Proton   Impact   Transition
C                           'H' => Charge   Exchange Recombination
C                           'R' => Free     Electron Recombination
C                           'I' => Electron Impact Ionisation
C  INPUT : (I*4)  I1A()   = TRANSITION:
C                            LOWER ENERGY LEVEL INDEX (CASE ' ' & 'P')
C                            NOT USED                 (CASE 'H' & 'R')
C  INPUT : (I*4)  I2A()   = TRANSITION:
C                            UPPER ENERGY LEVEL INDEX (CASE ' ' & 'P')
C                            CAPTURING    LEVEL INDEX (CASE 'H' & 'R')
C  INPUT : (R*8)  AVAL()  = TRANSITION:
C                            A-VALUE (SEC-1)          (CASE ' ')
C                            NEUTRAL BEAM ENERGY      (CASE 'H')
C                            NOT USED                 (CASE 'P' & 'R')
C
C  OUTPUT: (I*4)  ICNTE   = NUMBER OF ELECTRON IMPACT TRANSITIONS INPUT
C  OUTPUT: (I*4)  ICNTP   = NUMBER OF PROTON IMPACT TRANSITIONS INPUT
C  OUTPUT: (I*4)  ICNTR   = NUMBER OF FREE ELECTRON RECOMBINATIONS INPUT
C  OUTPUT: (I*4)  ICNTH   = NO. OF CHARGE EXCHANGE RECOMBINATIONS INPUT
C  OUTPUT: (I*4)  ICNTI   = NO. OF INNNER SHELL IONISATION INPUT
```

```
C
C  OUTPUT: (I*4)  IETRN() = INDEX VALUES IN MAIN TRANSITION ARRAYS WHICH
C                           1ST. DIM.: EL-TRANS. INDEX
C                           REPRESENT ELECTRON IMPACT TRANSITIONS.
C  OUTPUT: (C*1)  PECODE()= ELECTRONIC TRANSITION PLOT SELECTOR:
C                           ' ' => do not plot
C                           'P' or 'p' => plot
C                           1ST. DIM.: EL-TRANS. INDEX
C  OUTPUT: (C*1)  TECODE()= ELECTRONIC TRANSITION: DATA TYPE POINTER:
C                           ' ' => unassigned
C                           '1' => dipole
C                           '2' => non-dipole, non-spin change
C                           '3' => spin change
C                           '4' => small oscillator strength
C                           1ST. DIM.: EL-TRANS. INDEX
C
C  OUTPUT: (I*4)  IE1A()  = EL-TRANS. LOWER ENERGY LEVEL INDEX
C                           1ST. DIM.: EL-TRANS. INDEX
C  OUTPUT: (I*4)  IE2A()  = EL-TRANS. UPPER ENERGY LEVEL INDEX
C                           1ST. DIM.: EL-TRANS. INDEX
C  OUTPUT: (R*8)  AA()    = EL-TRANS.  A-VALUE (SEC-1)
C                           1ST. DIM.: EL-TRANS. INDEX
C  OUTPUT: (R*8)  CEA()   = EL-TRANS. BURGESS & TULLY C-VALUE
C                           1ST. DIM.: EL-TRANS. INDEX
C
C
C          (R*8)  CEREF   = PARAMETER = REFERENCE VALUE FOR B&T C-VAL.
C          (R*8)  FZERO   = PARAMETER = EFF. ZERO FOR F-VALUES IN
C                                       BURGESS & TULLY TYPE SELECTION.
C          (R*8)  FBIG    = PARAMETER = F-VALUE FOR TYPE SWITCH 1-4 IN
C                                       BURGESS & TULLY TYPE SELECTION.
C
C
C ROUTINES: NONE
C
C AUTHOR:  HP SUMMERS, UNIVERSITY OF STRATHCLYDE
C          JA8.08
C          TEL.  0141-553-4196
C
C DATE  :  04/06/98
C
C UPDATE:
C
C VERSION:  1.1 DATE: 09/08/98
C MODIFIED: RICHARD MARTIN
C - PUT UNDER SCCS CONTROL.
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      CHARACTER*(*)       CSTRGA(NDLEV)
      CHARACTER           PECODE(NDTRN),            TCODE(NDTRN)
      CHARACTER           TECODE(NDTRN)
      INTEGER             I1A(NDTRN),  I2A(NDTRN),  IA(NDLEV),   ICNTE
      INTEGER             ICNTH,       ICNTI,       ICNTP,       ICNTR
      INTEGER             IE1A(NDTRN), IE2A(NDTRN), IETRN(NDTRN)
      INTEGER             IL,          ILA(NDLEV),  ISA(NDLEV),  ITRAN
      INTEGER             IZ1,         NDLEV,       NDTRN
      REAL*8              AA(NDTRN),   AVAL(NDTRN), CEA(NDTRN)
      REAL*8              WA(NDLEV),   XJA(NDLEV)
```

3.85 bfwr11: Subroutine bfwr11 from library adas2xx

```
C
       SUBROUTINE BFWR11( IUNT11 , DSNINP ,
     &                    NDMET  , NDLEV  , NDTRN , NVMAX ,
     &                    DATE   ,
     &                    TITLED , IZ     , IZ0   , IZ1   , BWNO  ,
     &                    NPL    , BWNOA  , LBSETA, PRTWTA, CPRTA ,
     &                    IL     ,
     &                    IA     , CSTRGA , ISA   , ILA   , XJA   , WA ,
     &                    CPLA   , NPLA   , IPLA  , ZPLA  ,
     &                    CIONP  ,
     &                    NVN    , SCEFN  ,
     &                    ITRAN  , MAXLEV ,
     &                    TCODE  , I1A    , I2A   , AVAL  , SCOMN
     &                  )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: BFWR11 *********************
C
C  PURPOSE:  PRODUCES AN ADF04 TYPE FILE, WHERE THE CONTENTS IS
C            CONSIDERED AS THE OUTPUT DATA SET FROM ADAS215.
C
C  CALLING PROGRAM: ADAS215
C
C
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  IUNT11  = UNIT TO WHICH INPUT FILE IS ALLOCATED
C  INPUT : (C*80) DSNINP  = NAME OF INPUT ADF04 FILE
C  INPUT : (I*4)  NDMET   = MAXIMUM NUMBER OF PARENTS
C  INPUT : (I*4)  NDLEV   = MAXIMUM NUMBER OF LEVELS
C  INPUT : (I*4)  NDTRN   = MAX. NUMBER OF TRANSITIONS
C  INPUT : (I*4)  NVMAX   = MAX. NUMBER OF TEMPERATURES
C  INPUT : (C*8)  DATE    = DATE (AS DD/MM/YY).
C  INPUT : (C*10) USERID  = USER IDENTIFIER OF CODE EXECUTOR.
C  INPUT : (C*3)  TITLED  = ELEMENT SYMBOL.
C  INPUT : (I*4)  IZ      =  RECOMBINED ION CHARGE READ
C  INPUT : (I*4)  IZ0     =         NUCLEAR CHARGE READ
C  INPUT : (I*4)  IZ1     = RECOMBINING ION CHARGE READ
C                           (NOTE: IZ1 SHOULD EQUAL IZ+1)
C  INPUT : (R*8)  BWNO    = IONISATION POTENTIAL (CM-1)
C  INPUT : (I*4)  NPL     = NO. OF PARENTS ON FIRST LINE OF ADF04 FILE
C  INPUT : (R*8)  BWNOA() = IONISATION POTENTIAL (CM-1) OF PARENTS
C                           1ST.DIM.: PARENT INDEX
C  INPUT : (R*8)  PRTWTA()= PARENT WEIGHT FOR BWNOA()
C                           1ST.DIM.: PARENT INDEX
C  INPUT : (C*9)  CPRTA()  = PARENT NAME IN BRACKETS
C                            1ST DIM.: PARENT INDEX
C  INPUT : (L*4)  LBSETA() = .TRUE.  - PARENT WEIGHT SET FOR BWNOA()
C                            .FALSE. - PARENT WEIGHT NOT SET FOR BWNOA()
C
C  INPUT : (I*4)  IL      = INPUT DATA FILE: NUMBER OF ENERGY LEVELS
C
C  INPUT : (I*4)  IA()    = ENERGY LEVEL INDEX NUMBER
C  INPUT : (C*18) CSTRGA()= NOMENCLATURE/CONFIGURATION FOR LEVEL 'IA()'
C  INPUT : (I*4)  ISA()   = MULTIPLICITY FOR LEVEL 'IA()'
C                           NOTE: (ISA-1)/2 = QUANTUM NUMBER (S)
C  INPUT : (I*4)  ILA()   = QUANTUM NUMBER (L) FOR LEVEL 'IA()'
C  INPUT : (R*8)  XJA()   = QUANTUM NUMBER (J-VALUE) FOR LEVEL 'IA()'
```

```
C                             NOTE: (2*XJA)+1 = STATISTICAL WEIGHT
C  INPUT : (R*8)  WA()     = ENERGY RELATIVE TO LEVEL 1 (CM-1) FOR LEVEL
C                             'IA()'
C  INPUT : (C*1)  CPLA()   = CHAR. SPECIFYING 1ST PARENT FOR LEVEL 'IA()'
C                             INTEGER - PARENT IN BWNOA() LIST
C                             'BLANK' - PARENT BWNOA(1)
C                               'X'   - DO NOT ASSIGN A PARENT
C                             1ST DIM.: LEVEL INDEX
C  INPUT : (I*4)  NPLA()   = NO. OF PARENT/ZETA CONTRIBUTIONS TO IONIS.
C                             OF LEVEL
C                             1ST DIM.: PARENT INDEX
C  INPUT : (I*4)  IPLA(,)  = PARENT INDEX FOR CONTRIBUTIONS TO IONIS.
C                             OF LEVEL
C                             1ST DIM.: PARENT INDEX
C                             2ND DIM.: LEVEL INDEX
C  INPUT : (R*8)  ZPLA(,   = EFF. ZETA PARAM. FOR CONTRIBUTIONS TO IONIS.
C                             OF LEVEL
C                             1ST DIM.: PARENT INDEX
C  INPUT : (C*92) CIONP    = STRING CONTAINING LEVEL TERMINATOR AND
C                             IONISATION POTENTIALS
C
C  INPUT : (I*4)  NVN      = INPUT DATA FILE: NUMBER OF GAMMA/TEMPERATURE
C                             PAIRS FOR A GIVEN TRANSITION.
C  INPUT : (R*8)  SCEFN()  = INPUT DATA FILE: ELECTRON TEMPERATURES (K)
C                             (INITIALLY JUST THE MANTISSA. SEE 'ITPOW()')
C                             (NOTE: TE=TP=TH IS ASSUMED)
C
C  INPUT : (I*4)  ITRAN    = INPUT DATA FILE: NUMBER OF TRANSITIONS
C  INPUT : (I*4)  MAXLEV   = HIGHEST INDEX LEVEL IN READ TRANSITIONS
C
C  INPUT : (C*1)  TCODE()  = TRANSITION: DATA TYPE POINTER:
C                             ' ' => Electron Impact   Transition
C                             'P' => Proton   Impact   Transition
C                             'H' => Charge   Exchange Recombination
C                             'R' => Free     Electron Recombination
C  INPUT : (I*4)  I1A()    = TRANSITION:
C                              LOWER ENERGY LEVEL INDEX (CASE ' ' & 'P')
C                              SIGNED PARENT      INDEX (CASE 'H' & 'R')
C  INPUT : (I*4)  I2A()    = TRANSITION:
C                              UPPER ENERGY LEVEL INDEX (CASE ' ' & 'P')
C                              CAPTURING    LEVEL INDEX (CASE 'H' & 'R')
C  INPUT : (R*8)  AVAL()   = TRANSITION:
C                              A-VALUE (SEC-1)          (CASE ' ')
C                              NEUTRAL BEAM ENERGY      (CASE 'H')
C                              NOT USED                 (CASE 'P' & 'R')
C  INPUT : (R*8)  SCOMN(,) = TRANSITION:
C                              GAMMA VALUES             (CASE ' ' & 'P')
C                              RATE COEFFT. (CM3 SEC-1) (CASE 'H' & 'R')
C                             1ST DIMENSION - TEMPERATURE 'SCEF()'
C                             2ND DIMENSION - TRANSITION NUMBER
C
C
C  ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4UNIT     ADAS      FETCH UNIT NUMBER FOR OUTPUT OF MESSAGES
C          XXNAME     ADAS      FINDS REAL NAME OF USER
C          XXWSTR     ADAS      WRITES STRING TO A UNIT WITH TRAILING
C                               BLANKS REMOVED
C
C
```

```
C  AUTHOR : H. P. SUMMERS, UNIVERSITY OF STRATHCLYDE
C           JA8.08
C           TEL.  0141-553-4196
C
C  DATE:    04/06/96
C
C  UPDATE:
C
C VERSION:  1.1 DATE: 09/08/98
C MODIFIED: RICHARD MARTIN
C - PUT UNDER SCCS CONTROL.
C
C VERSION:  1.2 DATE: 15/04/99
C MODIFIED: Martin O'Mullane
C - Add real name of user via XXNAME.
C - Remove trailing blanks from adf04 file.
C
C-----------------------------------------------------------------------
      CHARACTER*92        CIONP
      CHARACTER           CPLA(NDLEV)
      CHARACTER*9         CPRTA(NDMET)
      CHARACTER*18        CSTRGA(NDLEV)
      CHARACTER*8         DATE
      CHARACTER*80        DSNINP
      CHARACTER           TCODE(NDTRN)
      CHARACTER*3         TITLED
      INTEGER             I1A(NDTRN),  I2A(NDTRN),  IA(NDLEV),    IL
      INTEGER             ILA(NDLEV),  IPLA(NDMET,NDLEV)
      INTEGER             ISA(NDLEV),  ITRAN,       IUNT11,       IZ
      INTEGER             IZ0,         IZ1,         MAXLEV,       NDLEV
      INTEGER             NDMET,       NDTRN,       NPL
      INTEGER             NPLA(NDLEV), NVMAX,       NVN
      LOGICAL             LBSETA(NDMET)
      REAL*8              AVAL(NDTRN), BWNO,        BWNOA(NDMET)
      REAL*8              PRTWTA(NDMET),            SCEFN(NVMAX)
      REAL*8              SCOMN(NVMAX,NDTRN),       WA(NDLEV)
      REAL*8              XJA(NDLEV),  ZPLA(NDMET,NDLEV)
```

3.86 bgcoef: Subroutine bgcoef from library adas2xx

```
C
       SUBROUTINE BGCOEF(LPSEL  , LZSEL  , LIOSEL ,
     &                   LHSEL  , LRSEL  , LISEL  , LNSEL ,
     &                   NMET   , IMETR  , IFOUT  , IDOUT ,
     &                   MAXT   , TINE   , TINP   , TINH  ,
     &                   MAXD   , DINE   , DINP   ,
     &                   AA     , XIA    , XJA    , ER    , ZEFF  ,
     &                   ICNTE  , ICNTP  , ICNTR  , ICNTH , ICNTI ,
     &                   IETRN  , IPTRN  , IRTRN  , IHTRN , IITRN ,
     &                   IE1A   , IE2A   , IP1A   , IP2A  ,
     &                   I1A    , I2A    ,
     &                   IL     , NV     , SCEF   , TSCEF , SCOM  ,
     &                   IUNT27 , OPEN27 , IZ0    , IZ1   ,
     &                   DSNEXP , DSNINC ,
     &                   MAXLEV , npl    , nplr   , npli,NPLA   , IPLA ,
     &                   BWNO   , BWNOA  , PRTWTA ,
     &                   POPAR
     &                  )



C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: BGCOEF *********************
C
C  PURPOSE: CALCULATES COLLISIONAL-RADIATIVE POPULATION
C
C  CALLING PROGRAM: ADAS216
C
C  ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C
C  THIS IS A SUBROUTINE VERSION OF ADAS208 WITHOUT THE SEARCH FOR IONISATION
C  RATES FROM ADF07 FILES.
C
C  AUTHOR   : Martin O'Mullane,
C             K1/1/43,
C             JET
C
C  VERSION  : 1.1
C  DATE     : 17/03/1999
C
C  MODIFIED : Martin O'Mullane
C             First version.
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      CHARACTER*80        DSNEXP,      DSNINC
      INTEGER             I1A(NDTRN),  I2A(NDTRN),  ICNTE,       ICNTH
      INTEGER             ICNTI,       ICNTP,       ICNTR,       IDOUT
      INTEGER             IE1A(NDTRN), IE2A(NDTRN), IETRN(NDTRN)
      INTEGER             IFOUT,       IHTRN(NDTRN)
      INTEGER             IITRN(NDTRN),             IL
      INTEGER             IMETR(NDMET),             IP1A(NDTRN)
      INTEGER             IP2A(NDTRN), IPLA(NDMET,NDLEV)
      INTEGER             IPTRN(NDTRN),             IRTRN(NDTRN)
      INTEGER             IUNT27,      IZ0,         IZ1,         MAXD
      INTEGER             MAXLEV,      MAXT,        NMET,        NPL
```

```
INTEGER            NPLA(NDLEV), NPLI,         NPLR,         NV
LOGICAL            LHSEL,       LIOSEL,       LISEL,        LNSEL
LOGICAL            LPSEL,       LRSEL,        LZSEL,        OPEN27
REAL*8             AA(NDTRN),   BWNO,         BWNOA(NDMET)
REAL*8             DINE(NDDEN), DINP(NDDEN), ER(NDLEV)
REAL*8             POPAR(NDLEV,NDTEM,NDDEN), PRTWTA(NDMET)
REAL*8             SCEF(14),    SCOM(14,NDTRN)
REAL*8             TINE(NDTEM), TINH(NDTEM), TINP(NDTEM)
REAL*8             TSCEF(14,3), XIA(NDLEV),  XJA(NDLEV),  ZEFF
```

3.87 bgdiff: Subroutine bgdiff from library adas2xx

```
C
      SUBROUTINE BGDIFF( ILEV  , maxt  , maxd    ,
     &                   popun , popar ,
     &                   error , index ,
     &                   ind_t , err_t , index_t , adiff_t , rt ,
     &                   ind_d , err_d , index_d , adiff_d , rd
     &                 )

C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: BGDIFF *********************
C
C  PURPOSE:  Calculates the absolute difference from 1.0 for temperature
C            or density and updates the set of arrays holding the top
C            ndtr contributing transitions.
C
C  CALLING PROGRAM: ADAS216
C
C  INPUT : (R*8)   POP      = POPULATION ARRAY
C
C  ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          R8ADIF     ADAS      calculates absolute difference of array
C
C
C  AUTHOR   : Martin O'Mullane,
C             K1/1/43,
C             JET
C
C  VERSION  : 1.1
C  DATE     : 17/03/1999
C
C  MODIFIED : Martin O'Mullane
C             First version.
C
C-----------------------------------------------------------------------
      INTEGER             ILEV,         INDEX
      INTEGER             INDEX_D(NDLEV,NDTEM,NDTR)
      INTEGER             INDEX_T(NDLEV,NDDEN,NDTR)
      INTEGER             IND_D(NDLEV,NDTEM),        IND_T(NDLEV,NDDEN)
      INTEGER             MAXD,         MAXT
      REAL*8              ADIFF_D(NDLEV,NDTEM,NDTR)
      REAL*8              ADIFF_T(NDLEV,NDDEN,NDTR),              ERROR
      REAL*8              ERR_D(NDLEV,NDTEM,NDTR)
      REAL*8              ERR_T(NDLEV,NDDEN,NDTR)
      REAL*8              POPAR(NDLEV,NDTEM,NDDEN)
      REAL*8              POPUN(NDLEV,NDTEM,NDDEN)
      REAL*8              RD(NDLEV,NDTEM,NDDEN,NDTR)
      REAL*8              RT(NDLEV,NDTEM,NDDEN,NDTR)
```

3.88 bgeset: Subroutine bgeset from library adas2xx

```
C
      SUBROUTINE BGESET( NDTRN  , NDGEN  , NDSPF  , NPLR   , NPLI  ,
     &                   ITRAN  , TCODE  ,
     &                   ICNTE  , ICNTP  , ICNTR  , ICNTH  , ICNTI ,
     &                   IETRN  , IPTRN  , IRTRN  , IHTRN  , IITRN ,
     &                   IE1A   , IE2A   , IP1A   , IP2A   ,
     &                   NUMEXC , DEFEXC , GENEXC , SPFEXC ,
     &                   NUMREC , DEFREC , GENREC , SPFREC ,
     &                   NUMCXR , DEFCXR , GENCXR , SPFCXR ,
     &                   NUMION , DEFION , GENION , SPFION ,
     &                   ERRMUL
     &                  )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: BGESET *********************
C
C  PURPOSE:  TO SET THE ERROR MULTIPLER ACCORDING TO ERROR VALUES
C
C  CALLING PROGRAM: ADAS216
C
C  ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C
C
C  AUTHOR   : Martin O'Mullane,
C             K1/1/43,
C             JET
C
C  VERSION  : 1.1
C  DATE     : 17/03/1999
C
C  MODIFIED : Martin O'Mullane
C             First version.
C
C-----------------------------------------------------------------------
      CHARACTER           TCODE(NDTRN)
      INTEGER             ICNTE,       ICNTH,       ICNTI,       ICNTP
      INTEGER             ICNTR,       IE1A(NDTRN), IE2A(NDTRN)
      INTEGER             IETRN(NDTRN),             IHTRN(NDTRN)
      INTEGER             IITRN(NDTRN),             IP1A(NDTRN)
      INTEGER             IP2A(NDTRN), IPTRN(NDTRN)
      INTEGER             IRTRN(NDTRN),             ITRAN,       NDGEN
      INTEGER             NDSPF,       NDTRN,       NPLI,        NPLR
      INTEGER             NUMCXR(3),   NUMEXC(3),   NUMION(3)
      INTEGER             NUMREC(3)
      REAL*8              DEFCXR,      DEFEXC,      DEFION,      DEFREC
      REAL*8              ERRMUL(NDTRN),            GENCXR(NDGEN,2)
      REAL*8              GENEXC(NDGEN,2),          GENION(NDGEN,2)
      REAL*8              GENREC(NDGEN,2),          SPFCXR(NDSPF,3)
      REAL*8              SPFEXC(NDSPF,3),          SPFION(NDSPF,3)
      REAL*8              SPFREC(NDSPF,3)
```

3.89 bghist: Subroutine bghist from library adas2xx

```
C
      SUBROUTINE BGHIST( IUNIT , ILEV  , ITE   , IDEN  ,
     &                   IHIST , ih_out
     &                 )

C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: BGHIST *********************
C
C  PURPOSE:  WRITES UNFORMATTED HISTOGRAM DATA TO IUNIT.
C
C  CALLING PROGRAM: ADAS216
C
C  INPUT : (I*4)   IUNIT    = UNIT NO. OF OPENED UNFORMATTED FILE
C  INPUT : (R*8)   POP      = POPULATION ARRAY
C
C  ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C
C
C  AUTHOR   : Martin O'Mullane,
C             K1/1/43,
C             JET
C
C  VERSION  : 1.1
C  DATE     : 17/03/1999
C
C  MODIFIED : Martin O'Mullane
C             First version.
C
C-----------------------------------------------------------------------
       INTEGER             IDEN,         IHIST(NDLEV,NDTEM,NDDEN,NDBIN)
       INTEGER*2           IH_OUT(ILEV,ITE,IDEN,NDBIN)
       INTEGER             ILEV,         ITE,          IUNIT
```

3.90 bgperr: Subroutine bgperr from library adas2xx

```
C
      SUBROUTINE BGPERR( NDGEN  , NDSPF   ,
     &                   NUM    , ERRARR  ,
     &                   NARR   , DEF     , GEN     , SPF  )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: BGRERR *********************
C
C  PURPOSE: PARSES ERROR INFORMATION FROM ADF04 FILES.
C
C  CALLING PROGRAM: ADAS216
C
C  SUBROUTINE:
C
C  INPUT : (I*4)   NDGEN    = MAX. NUMBER OF GENERAL ERRORS ALLOWED
C  INPUT : (I*4)   NDSPF    = MAX. NUMBER OF SPECIFIC ERRORS ALLOWED
C
C  INPUT:  (I*4)   NUM      = NUMBER OF ERROR LINES
C  INPUT:  (C*70)  ERRARR() = COMPRESSED STRING OF ERRORS
C
C
C  OUTPUT: (R*8)   NARR(3)  = NO. OF ERROR TYPES FOR EXCITATION
C                               1ST INDEX:  NO. DEFAULT (0 OR 1)
C                               2ND INDEX:  NO. GENERAL
C                               3RD INDEX:  NO. SPECIFIC
C  OUTPUT: (R*8)   DEF      = DEFAULT ERROR
C  OUTPUT: (R*8)   GEN(,)   = GENERAL ERROR
C                               1ST DIMENSION: SOURCE/DESTINATION LEVEL
C                               2ND DIMENSION: ERROR
C
C  OUTPUT: (R*8)   SPF(,)   = SPECIFIC ERROR
C                               1ST DIMENSION: FIRST LEVEL
C                               2ND DIMENSION: SECOND LEVEL
C                               3RD DIMENSION: ERROR
C
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          XXFCHR     ADAS      GETS OCCURANCES OF SUBSTRING IN A STRING
C          I4UNIT     ADAS      FETCH UNIT NUMBER FOR OUTPUT OF MESSAGES
C
C
C  AUTHOR  : MARTIN O'MULLANE,
C            K1/1/43,
C            JET
C
C VERSION  : 1.1                          DATE: 17/03/1999
C MODIFIED : MARTIN O'MULLANE
C            FIRST VERSION.
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      CHARACTER*70        ERRARR(1+NDGEN+NDSPF)
      INTEGER             NARR(3),     NDGEN,       NDSPF,       NUM
      REAL*8              DEF,         GEN(NDGEN,2)
      REAL*8              SPF(NDSPF,3)
```

3.91 bgrerr: Subroutine bgrerr from library adas2xx

```
C
      SUBROUTINE BGRERR( IUNT   ,
     &                   NUMEXC , DEFEXC  , GENEXC  , SPFEXC  ,
     &                   NUMREC , DEFREC  , GENREC  , SPFREC  ,
     &                   NUMCXR , DEFCXR  , GENCXR  , SPFCXR  ,
     &                   NUMION , DEFION  , GENION  , SPFION  )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: BGRERR *********************
C
C  PURPOSE: READS ERROR INFORMATION IN ADF04 SPECIFIC ION FILE AND
C           PASSES BACK 4 ARRAYS ASSOCIATED WITH EACH TRANSITION TPYE
C           OF EXCITATION, RECOMBINATION AND CHARGE EXCHANGE.
C
C           THERE ARE 3 TYPES OF ERROR FOR EACH TYPE, NAMELY
C           DEFAULT,  GENERAL OR SPECIFIC.
C            - GENERAL IS ERROR FROM/TO A PARTICULAR LEVEL (LIMITED TO 5).
C            - SPECIFIC IS THE ERROR TO BE APPLIED TO ONE TRANSITION
C              (LIMITED TO 20).
C            - DEFAULT IS FOR THE REST (ONLY 1!).
C           ERRORS ARE GIVEN AS A PERCENTAGE INTEPRETTATED AS VALUE +/-
C           THE PERCENTAGE ERROR.
C           ERRORS ARE IN FREE-FORMAT MARKED BY LINES IN ADF04 FILE. E.G.
C           FOR EXCITATION AND RECOMBINATION SPECIFICATION
C
C            C+++ERROR
C            C Excitation
C            C
C            C    *-*  4.7
C            C     3-*  3.2
C            C     *- 4  3.2
C            C     1-*  10.3
C            C      1- 2  5.4
C            C      31- 2  7.4
C            C      31- 4  7.4
C            C Recombination
C            C
C            C  * -*   20.0
C            C  *-1  30.0
C            C+++ERROR
C
C           HERE *-* REPRESENTS DEFAULT, 1-* OR *-3 REPRESENTS GENERAL, 3-5
C           DEFINES ERROR ON SPECIFIC TRANSITION. CAPITALISATION OR SPACING
C           IS NOT IMPORTANT.
C
C  CALLING PROGRAM: ADAS216
C
C  SUBROUTINE:
C
C  PARAMETER: (I*4)   NDGEN    = MAX. NUMBER OF GENERAL ERRORS ALLOWED
C  PARAMETER: (I*4)   NDSPF    = MAX. NUMBER OF SPECIFIC ERRORS ALLOWED
C
C  INPUT : (I*4)   IUNT     = UNIT NO. OF OPENED ADF04 FILE
C
C  OUTPUT: (R*8)   NUMEXC(3)= NO. OF ERROR TYPES FOR EXCITATION
C                             1ST INDEX:  NO. DEFAULT (0 OR 1)
C                             2ND INDEX:  NO. GENERAL
C                             3RD INDEX:  NO. SPECIFIC
C  OUTPUT: (R*8)   DEFEXC   = DEFAULT ERROR
```

```
C  OUTPUT: (R*8)   GENEXC(,)= GENERAL ERROR
C                              1ST DIMENSION: SOURCE/DESTINATION LEVEL
C                              2ND DIMENSION: ERROR
C
C  OUTPUT: (R*8)   SPFEXC(,)= SPECIFIC ERROR
C                              1ST DIMENSION: FIRST LEVEL
C                              2ND DIMENSION: SECOND LEVEL
C                              3RD DIMENSION: ERROR
C
C NOTE : SIMILAR DEFINITIONS FOR RECOMBINATION, CHARGE EXCHANGE
C        AND IONISATION.
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          XXSTUC     ADAS      CONVERTS STRING TO UPPERCASE
C          XXRMWS     ADAS      REMOVES BLANKS FROM A STRING
C          BGPERR     ADAS      PARSES ERROR STRING
C
C
C  AUTHOR  : Martin O'Mullane,
C            K1/1/43,
C            JET
C
C VERSION  : 1.1
C DATE     : 17/03/1999
C MODIFIED : Martin O'Mullane
C            First version.
C
C VERSION  : 1.2
C DATE     : 03-01-2006
C MODIFIED : Martin O'Mullane
C              - Increase NDSPF to 2100.
C
C VERSION  : 1.3
C DATE     : 18-05-2007
C MODIFIED : Allan Whiteford
C              - Updated comments as part of subroutine documentation
C                procedure.
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IUNT,         NUMCXR(3),   NUMEXC(3)
      INTEGER             NUMION(3),    NUMREC(3)
      REAL*8              DEFCXR,       DEFEXC,      DEFION,      DEFREC
      REAL*8              GENCXR(NDGEN,2),           GENEXC(NDGEN,2)
      REAL*8              GENION(NDGEN,2),           GENREC(NDGEN,2)
      REAL*8              SPFCXR(NDSPF,3),           SPFEXC(NDSPF,3)
      REAL*8              SPFION(NDSPF,3),           SPFREC(NDSPF,3)
```

3.92 bgrwer: Subroutine bgrwer from library adas2xx

```
C
      SUBROUTINE BGRWER( NDGEN  , NDSPF   ,
     &                   IRW    ,
     &                   NARR   , DEF     , GEN     , SPF  )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: BGRERR *********************
C
C  PURPOSE: TO READ OR WRITE ERROR INFORMATION FROM/TO IDL. THE SIZE
C           OF THE ARRAYS HAS ALREADY BEEN PASSES TO IDL.
C
C  CALLING PROGRAM: ADAS216
C
C  SUBROUTINE:
C
C  INPUT : (I*4)   NDGEN    = MAX. NUMBER OF GENERAL ERRORS ALLOWED
C  INPUT : (I*4)   NDSPF    = MAX. NUMBER OF SPECIFIC ERRORS ALLOWED
C
C  INPUT:  (I*4)   irw      = switch for read/write to idl
C                             irw = 0 read
C                             irw = 1 write
C
C
C  OUTPUT: (R*8)   NARR(3)  = NO. OF ERROR TYPES FOR EXCITATION
C                                1ST INDEX:  NO. DEFAULT (0 OR 1)
C                                2ND INDEX:  NO. GENERAL
C                                3RD INDEX:  NO. SPECIFIC
C  OUTPUT: (R*8)   DEF      = DEFAULT ERROR
C  OUTPUT: (R*8)   GEN(,)   = GENERAL ERROR
C                                1ST DIMENSION: SOURCE/DESTINATION LEVEL
C                                2ND DIMENSION: ERROR
C
C  OUTPUT: (R*8)   SPF(,)   = SPECIFIC ERROR
C                                1ST DIMENSION: FIRST LEVEL
C                                2ND DIMENSION: SECOND LEVEL
C                                3RD DIMENSION: ERROR
C
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C
C
C  AUTHOR  : MARTIN O'MULLANE,
C            K1/1/43,
C            JET
C
C VERSION  : 1.1                          DATE: 17/03/1999
C MODIFIED : MARTIN O'MULLANE
C            FIRST VERSION.
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IRW,         NARR(3),     NDGEN,       NDSPF
      REAL*8              DEF,         GEN(NDGEN,2)
      REAL*8              SPF(NDSPF,3)
```

3.93 bgseed: Subroutine bgseed from library adas2xx

```
C
      SUBROUTINE BGSEED( ISEED )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: BGSEED *********************
C
C  PURPOSE: Sets iseed to a value based on the time/date. This seed
C           is for use with the LAPACK routine dlarnd.
C
C           The seed array of the random number generator
C           elements must be between 0 and 4095, and ISEED(4)
C           must be odd.
C
C  CALLING PROGRAM: ADAS216
C
C  SUBROUTINE:
C
C  I/O   : (I*4)   ISEED()  = 4 element seed
C
C
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C
C
C  AUTHOR  : MARTIN O'MULLANE,
C            K1/1/43,
C            JET
C
C VERSION  : 1.1                          DATE: 17/03/1999
C MODIFIED : MARTIN O'MULLANE
C            FIRST VERSION.
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             ISEED(4)
```

3.94 bgsetm: Subroutine bgsetm from library adas2xx

```
C
       SUBROUTINE BGSETM( IZ0    , IZ     ,
     &                    NDLEV  , IL     , ICNTE ,
     &                    CSTRGA , ISA    , ILA   , XJA  ,
     &                    STRGA  , NPL    , CPRTA , NDMET,
     &                    STRGMF , STRGMI
     &                  )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: BGSETM *********************
C
C  PURPOSE:  TO SET UP PARAMETERS TO PASS TO IDL
C
C  CALLING PROGRAM: ADAS216
C
C  DATA:
C           DATA IS OBTAINED VIA SUBROUTINE 'BADATA'
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  IZ0     = NUCLEAR CHARGE READ
C  INPUT : (I*4)  IZ      = RECOMBINED ION CHARGE READ
C
C  INPUT : (I*4)  NDLEV   = MAXIMUM NUMBER OF ENERGY LEVELS ALLOWED
C  INPUT : (I*4)  IL      = INPUT DATA FILE: NUMBER OF ENERGY LEVELS
C  INPUT : (I*4)  ICNTE   = NUMBER OF ELECTRON IMPACT TRANSITIONS
C
C  INPUT : (C*18) CSTRGA()= NOMENCLATURE/CONFIGURATION FOR LEVEL 'IA()'
C  INPUT : (I*4)  ISA()   = MULTIPLICITY FOR LEVEL 'IA()'
C                           NOTE: (ISA-1)/2 = QUANTUM NUMBER (S)
C  INPUT : (I*4)  ILA()   = QUANTUM NUMBER (L) FOR LEVEL 'IA()'
C  INPUT : (R*8)  XJA()   = QUANTUM NUMBER (J-VALUE) FOR LEVEL 'IA()'
C                           NOTE: (2*XJA)+1 = STATISTICAL WEIGHT
C  INPUT : (I*4)  NPL     = NUMBER OF PARENTS IN INPUT DATA SET
C  INPUT : (C*9)  CPRTA() = PARENT NAME FROM INPUT DATA SET
C  INPUT : (I*4)  NDMET   = MAX.NO.OF METASTABLES ALLOWED
C
C  OUTPUT: (C*22) STRGA() = LEVEL DESIGNATIONS
C  OUTPUT: (C*11) STRGMF()= INFORMATION STRINGS FOR IDL
C  OUTPUT: (C*12) STRGMI()= INFORMATION STRINGS FOR IDL
C
C          (C*8)  F6      = PARAMETER = 'VREPLACE'
C
C          (I*4)  ILEN    = LENGTH, IN BYTES, OF ISPF DIALOG VARIABLES
C          (I*4)  ILEV    = ARRAY COUNTER FOR LEVEL INDEX
C          (I*4)  J       = VALUE OF QUANTUM NUMBER L + 1
C          (I*4)  LFPOOL  = NO. OF LEVEL STRINGS SENT TO FUNCTION POOL
C
C          (C*2)  SZ0     = NUCLEAR CHARGE READ
C          (C*2)  SZ      = RECOMBINED ION CHARGE READ
C          (C*4)  SCNTE   = NUMBER OF ELECTRON IMPACT TRANSITIONS
C          (C*4)  SIL     = NUMBER OF ENERGY LEVELS
C          (C*1)  CONFIG()= QUANTUM NUMBER (L) LETTERS
C                           DIMENSION: QUANTUM NUMBER L + 1
C          (C*8)  CHA()   = FUNCTION POOL NAMES: CHARGE VALUES
C          (C*8)  CHB()   = FUNCTION POOL NAMES: LEVEL DESIGNATIONS <=99
C          (I*4)  PIPEOU  = PARAMETER = UNIT NUMBER FOR OUTPUT TO PIPE
C
C
```

```
C ROUTINES:
C           ROUTINE     SOURCE      BRIEF DESCRIPTION
C           ------------------------------------------------------------
C
C BASED ON B8SETP.FOR
C AUTHOR:  D.H.BROOKS, UNIVERSITY OF STRATHCLYDE,  11/04/96
C
C
C AUTHOR  : Martin O'Mullane,
C            K1/1/43,
C            JET
C
C VERSION : 1.1                          DATE: 17/03/1999
C MODIFIED: Martin O'Mullane
C           First version.
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      CHARACTER*9         CPRTA(NDMET)
      CHARACTER*18        CSTRGA(IL)
      CHARACTER*22        STRGA(NDLEV)
      CHARACTER*11        STRGMF(NDMET)
      CHARACTER*12        STRGMI(NDLEV)
      INTEGER             ICNTE,       IL,          ILA(IL),     ISA(IL)
      INTEGER             IZ,          IZ0,         NDLEV,       NDMET
      INTEGER             NPL
      REAL*8              XJA(IL)
```

3.95 bgtest: Subroutine bgtest from library adas2xx

```
C
      SUBROUTINE BGTEST( IZ1    , IL      ,
     &                   IA     , ISA     , ILA   , XJA   , WA   ,
     &                   NV     , SCEF    ,
     &                   ITRAN  ,
     &                   TCODE  , I1A     , I2A   , AVAL  , SCOM ,
     &                   numcom , outcom
     &                 )

C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: BGTEST *********************
C
C  PURPOSE:   Examines the collison strengths of adf04 files for any
C             discrepencies, outlyiny points, mistakes etc.
C
C             Three methods are used for checking
C               o  fit a minmax polynominal and flag excessive errors
C               o  find large deviations from a 3-point running average
C               o  find excessive changes in slope
C
C             The first method really checks for smoothness with the
C             second looking for outlying points. (The minmax fit
C             should also spot these). The third is not so successful
C             and care should be taken in using it.
C
C             All collisions and temperatures are transformed by the
C             Burgess-Tully method before the tests are applied.
C
C             Based on off-line test_adf04.for code (Martin O'Mullane,
C             16-2-99).
C
C
C  CALLING PROGRAM: ADAS216
C
C  SUBROUTINE:
C
C  INPUT : (I*4)
C
C  ROUTINES:
C           ROUTINE    SOURCE    BRIEF DESCRIPTION
C           ------------------------------------------------------------
C           BGTRAN     ADAS      RETURNS BURGESS-TULLY TRANSFORMED
C                                TEMPERATURE AND COLLISION STRENGTH
C           I4UNIT     ADAS      FETCH UNIT NUMBER FOR OUTPUT OF MESSAGES
C
C
C   AUTHOR  : MARTIN O'MULLANE,
C             K1/1/43,
C             JET
C
C  VERSION  : 1.1                          DATE: 17/03/1999
C  MODIFIED : MARTIN O'MULLANE
C             FIRST VERSION.
C
C  VERSION  : 1.2                          DATE: 16/11/2001
C  MODIFIED : Martin O'Mullane
C             Problem with comment array being overwritten. Add check
C             but continue to process.
```

```
C
C-----------------------------------------------------------------------
      CHARACTER*80        OUTCOM(210+200*NVMAX)
      CHARACTER           TCODE(NDTRN)
      INTEGER             I1A(NDTRN),  I2A(NDTRN),  IA(NDLEV),    IL
      INTEGER             ILA(NDLEV),  ISA(NDLEV),  ITRAN,        IZ1
      INTEGER             NUMCOM,      NV
      REAL*8              AVAL(NDTRN), SCEF(NVMAX), SCOM(NVMAX,NDTRN)
      REAL*8              WA(NDLEV),   XJA(NDLEV)
```

3.96 bgtran: Subroutine bgtran from library adas2xx

```
C
      SUBROUTINE  BGTRAN( TYP   , C      ,
     &                    AIN   , WVNOU  , WVNOL , WTU   , WTL  ,
     &                    TEIN  , UPSIN  , NV    ,
     &                    x     , y
     &                  )

C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: BGTRAN *********************
C
C This subroutine is based on ADAS215's bftran.for - it is modified to
C output the Burgess and Tully x and y vectors.
C
C PURPOSE : TO IMPLEMENT THE TRANSFORMATION DESCRIBED BY
C           BURGESS AND TULLY ( SEE REFERENCE (1)) WHICH
C           IS USED TO ASSESS AND COMPACT DATA.
C
C
C REFERENCES:
C           (1) A.BURGESS AND J.A.TULLY
C           ON THE ANALYSIS OF COLLISION STRENGTHS
C           AND RATE COEFFICIENTS.
C           ASTRON.ASTROPHYS.254,436-453 (1992 )
C
C           (2) SUMMERS.H.P
C               ADAS USERS MANUAL ( 1ST EDITION ).
C
C INPUT :
C       (R*8) Z1          = THE ION CHARGE +1.
C       (C*1) TYP         = BURGESS & TULLY TRANSITION TYPE CODE
C       (R*8) C           = THE ADJUSTABLE PARAMETER ASSOCIATED
C                           WITH THE BURGESS AND TULLY
C                           TRANSFORMATION ( SEE REFERENCE (1)).
C       (R*8) AIN         = THE EINSTEIN 'A' CO-EFFICIENT. THIS
C                           IS READ DIRECTLY FROM THE ADF04
C                           TYPE FILE.
C       (R*8) WVNOU       = THE WAVENUMBER OF THE UPPER LEVEL.
C                           THIS IS READ DIRECTLY FROM THE ADF04
C                           TYPE FILE.
C       (R*8) WVNLO       = THE WAVENUMBER OF THE LOWER LEVEL.
C                           THIS IS READ DIRECTLY FROM THE ADF04
C                           TYPE FILE.
C       (R*8) WTU         = THE STATISTICAL WEIGHT OF THE UPPER
C                           LEVEL. THIS IS OBTAINED BY
C
C       (R*8) WTL         = THE STATISTICAL WEIGHT OF THE LOWER
C                           LEVEL. THIS IS OBTAINED BY
C
C       (R*8) TEIN        = THE TEMPERATURE ARRAY (K). THIS
C                           DATA IS READ DIRECTLY FROM THE
C                           ADF04 TYPE FILE.
C       (R*8) UPSIN       = THE ARRAY CONTAING THE EFFECTIVE
C                           COLLISION STRENGTH. THIS DATA IS
C                           READ DIRECTLY FROM THE ADF04
C                           TYPE FILE.
C       (I*4) NV          = THE NUMBER OF TEMPERATURE/EFFECTIVE
C                           COLLISION STRENGTH PAIRS FOR A GIVEN
C                           TRANSITION.
```

```
C
C OUTPUT:
C
C       (R*8) X           = THE X ARRAY ASSOCIATED WITH THE
C                           BURGESS AND TULLY TRANSFORMATION.
C       (R*8) Y           = THE Y ARRAY ASSOCIATED WITH THE
C                           BURGESS AND TULLY TRANSFORMATION.
C
C
C       (R*8) E           = THE MATHEMATICAL CONSTANT E.
C       (R*8) CONST       = CLUSTER OF PHYSICAL CONSTANTS.
C                           SEE PAGE 12 OF REFERENCE (2).
C       (R*8) EIJIN       = THE TRANSITION ENERGY (RYD).
C       (R*8) FIJIN       = THE OSCILLATOR STRENGTH.
C       (R*8) ET          = GENERAL CONSTANT.
C       (R*8) C           = THE BURGESS C PARAMETER.
C       (R*8) DY          = DERIVATIVES AT INPUT KNOTS.
C                           SEE XXSPLN FOR FUTHER DETAILS.
C       (R*8) XOUT        = X ARRAY ASSOCIATED WITH THE
C                           BURGESS AND TULLY TRANSFORMATION.
C       (R*8) YOUT        = Y ARRAY ASSOCIATED WITH THE
C                           BURGESS AND TULLY TRANSFORMATION.
C       (I*4) NVMAX       = THE MAXIMUM NUMBER OF TEMPERTURES
C                           THAT CAN BE READ.
C       (I*4) IOPT        = GENERAL PARAMETER ASSOCIATED WITH
C                           THE SUBROUTINE XXSPLN.
C       (I*4) I           = GENERAL VARIABLE WHICH IS USED AS
C                           A COUNTER.
C       (LOG) LSETX       = PARAMETER ASSOCIATED WITH THE
C                           SUBROUTINE XXSPLN.
C
C AUTHOR:   H. P. SUMMERS, UNIVERSITY OF STRATHCLYDE
C           JA8.08
C           TEL.  0141-553-4196
C
C DATE:    04/06/98
C
C
C VERSION  : 1.1                          DATE: 17/03/1999
C MODIFIED : MARTIN O'MULLANE
C            FIRST VERSION.
C
C-----------------------------------------------------------------
      CHARACTER           TYP
      INTEGER             NV
      REAL*8              AIN,          C,            TEIN(NV)
      REAL*8              UPSIN(NV),    WTL,          WTU,          WVNOL
      REAL*8              WVNOU,        X(NVMAX),     Y(NVMAX)
```

3.97 bgwerr: Subroutine bgwerr from library adas2xx

```
C
      SUBROUTINE BGWERR( IUNIT  ,
     &                   NUMEXC , DEFEXC , GENEXC , SPFEXC ,
     &                   NUMREC , DEFREC , GENREC , SPFREC ,
     &                   NUMCXR , DEFCXR , GENCXR , SPFCXR ,
     &                   NUMION , DEFION , GENION , SPFION
     &                  )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: BGWERR *********************
C
C  PURPOSE:  Writes error information to IUNIT
C
C  CALLING PROGRAM: ADAS216
C
C  ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C
C
C  AUTHOR   : Martin O'Mullane,
C             K1/1/43,
C             JET
C
C  VERSION  : 1.1
C  DATE     : 17/03/1999
C
C  MODIFIED : Martin O'Mullane
C             First version.
C
C VERSION  : 1.2
C DATE     : 03-01-2006
C MODIFIED : Martin O'Mullane
C              - Increase NDSPF to 2100.
C
C-----------------------------------------------------------------------
      INTEGER             IUNIT,       NUMCXR(3),   NUMEXC(3)
      INTEGER             NUMION(3),   NUMREC(3)
      REAL*8              DEFCXR,      DEFEXC,      DEFION,      DEFREC
      REAL*8              GENCXR(NDGEN,2),          GENEXC(NDGEN,2)
      REAL*8              GENION(NDGEN,2),          GENREC(NDGEN,2)
      REAL*8              SPFCXR(NDSPF,3),          SPFEXC(NDSPF,3)
      REAL*8              SPFION(NDSPF,3),          SPFREC(NDSPF,3)
```

3.98 bgwexp: Subroutine bgwexp from library adas2xx

```
C
      SUBROUTINE BGWEXP( IUNIT  ,
     &                   NDLEV  , NDTEM  , NDDEN   , NDTR    ,
     &                   ILEV   , ITE    , IDEN    ,
     &                   ind_t  , err_t  , index_t , adiff_t , rt     ,
     &                   ind_d  , err_d  , index_d , adiff_d , rd     ,
     &                   int_out, it_out ,errt_out , adt_out , rt_out ,
     &                   ind_out, id_out ,errd_out , add_out , rd_out
     &                 )

C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: BGWEXP *********************
C
C  PURPOSE:  WRITES (UNFORMATTED) EXPLORE ERROR DATA TO IUNIT.
C
C  CALLING PROGRAM: ADAS216
C
C  INPUT : (I*4)   IUNIT    = UNIT NO. OF OPENED UNFORMATTED FILE
C
C  ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C
C
C  AUTHOR   : Martin O'Mullane,
C             K1/1/43,
C             JET
C
C  VERSION  : 1.1
C  DATE     : 17/03/1999
C
C  MODIFIED : Martin O'Mullane
C             First version.
C
C-----------------------------------------------------------------------
      INTEGER             IDEN,         ID_OUT(ILEV,ITE,NDTR),    ILEV
      INTEGER             INDEX_D(NDLEV,NDTEM,NDTR)
      INTEGER             INDEX_T(NDLEV,NDDEN,NDTR)
      INTEGER             IND_D(NDLEV,NDTEM),       IND_OUT(ILEV,ITE)
      INTEGER             IND_T(NDLEV,NDDEN),       INT_OUT(ILEV,IDEN)
      INTEGER             ITE,          IT_OUT(ILEV,IDEN,NDTR),   IUNIT
      INTEGER             NDDEN,        NDLEV,       NDTEM,       NDTR
      REAL                ADD_OUT(ILEV,ITE,NDTR)
      REAL*8              ADIFF_D(NDLEV,NDTEM,NDTR)
      REAL*8              ADIFF_T(NDLEV,NDDEN,NDTR)
      REAL                ADT_OUT(ILEV,IDEN,NDTR)
      REAL                ERRD_OUT(ILEV,ITE,NDTR)
      REAL                ERRT_OUT(ILEV,IDEN,NDTR)
      REAL*8              ERR_D(NDLEV,NDTEM,NDTR)
      REAL*8              ERR_T(NDLEV,NDDEN,NDTR)
      REAL*8              RD(NDLEV,NDTEM,NDDEN,NDTR)
      REAL                RD_OUT(ILEV,ITE,IDEN,NDTR)
      REAL*8              RT(NDLEV,NDTEM,NDDEN,NDTR)
      REAL                RT_OUT(ILEV,ITE,IDEN,NDTR)
```

3.99 bgwr33: Subroutine bgwr33 from library adas2xx

```
C
      SUBROUTINE BGWR33( iunit ,
     &                   ILEV  , ITE   , IDEN , pop , data_out
     &                 )

C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: BGWR33 *********************
C
C  PURPOSE:  WRITES UNFORMATTED DATA TO IUNIT
C
C  CALLING PROGRAM: ADAS216
C
C  INPUT : (I*4)   IUNIT    = UNIT NO. OF OPENED UNFORMATTED FILE
C  INPUT : (R*8)   POP      = POPULATION ARRAY
C
C  ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C
C
C  AUTHOR   : Martin O'Mullane,
C             K1/1/43,
C             JET
C
C  VERSION  : 1.1
C  DATE     : 17/03/1999
C
C  MODIFIED : Martin O'Mullane
C             First version.
C
C-----------------------------------------------------------------------
      INTEGER             IDEN,        ILEV,        ITE,          IUNIT
      REAL                DATA_OUT(ILEV,ITE,IDEN)
      REAL*8              POP(NDLEV,NDTEM,NDDEN)
```

3.100 bxchkm: Subroutine bxchkm from library adas2xx

```
      SUBROUTINE BXCHKM( NMET   , IMETR , ICNTE , IE1A , LMETR )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: BXCHKM *********************
C
C  PURPOSE: TO CHECK IF TRANSITIONS EXIST TO THE METASTABLE LEVELS.
C
C  CALLING PROGRAM: ADAS205/ADAS206
C
C  SUBROUTINE:
C
C  INPUT : (I*4)   NMET    = NUMBER OF METASTABLES ( 1 <= NMET <= 5 )
C  INPUT : (I*4)   IMETR() = INDEX OF METASTABLE IN COMPLETE LEVEL LIST
C  INPUT : (I*4)   ICNTE   = NUMBER OF ELECTRON IMPACT TRANSITIONS INPUT
C  INPUT : (I*4)   IE1A()  = ELECTRON IMPACT TRANSITION: LOWER ENERGY
C                            LEVEL INDEX.
C
C  OUTPUT: (L*4)   LMETR() = .TRUE.  =>ELECTRON IMPACT TRANSITION EXISTS
C                                      TO THE METASTABLE LEVEL  GIVEN BY
C                                       'IMETR()'.
C                            .FALSE. =>ELECTRON  IMPACT  TRANSITIONS  DO
C                                      NOT EXIST TO THE METASTABLE LEVEL
C                                      GIVEN BY 'IMETR()'.
C
C          (I*4)   I       = GENERAL USE
C          (I*4)   J       = GENERAL USE
C
C
C ROUTINES: NONE
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 4569
C
C DATE:    09/10/90
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             ICNTE,       IE1A(ICNTE), IMETR(NMET), NMET
      LOGICAL             LMETR(NMET)
```

3.101 bxcoef: Subroutine bxcoef from library adas2xx

```
      subroutine bxcoef(idlev  , idtrn  , idtem  , idden  , idmet ,
     &                  nmet   , imetr  , nord   , iordr  ,
     &                  maxt   , tine   , tinp   , tinh   , ifout ,
     &                  maxd   , dine   , dinp   , idout  ,
     &                  lpsel  , lzsel  , liosel ,
     &                  lhsel  , lrsel  , lisel  , lnsel  ,
     &                  iz     , iz0    , iz1    ,
     &                  npl    , bwno   , bwnoa  , prtwta ,
     &                  npla   , ipla   , zpla   , nplr   , npli  ,
     &                  il     , maxlev , xja    , wa     , zeff  ,
     &                  xia    , er     ,
     &                  icnte  , icntp  , icntr  , icnth  , icnti ,
     &                  icntl  , icnts  ,
     &                  ietrn  , iptrn  , irtrn  , ihtrn  , iitrn ,
     &                  iltrn  , istrn  ,
     &                  ie1a   , ie2a   , ip1a   , ip2a   , aa    ,
     &                  ia1a   , ia2a   , auga   ,
     &                  il1a   , il2a   , wvla   ,
     &                  is1a   , is2a   , lss04a ,
     &                  i1a    , i2a    ,
     &                  nv     , scef   , scom   ,
     &                  dsnexp , dsninc , iunt27 , open27 ,
     &                  stckm  , stvr   , stvi   , stvh   ,
     &                  stvrm  , stvim  , stvhm  , stack  ,
     &                  ltick
     &                 )



C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: BXCOEF *********************
C
C  PURPOSE: Calculates collisional-radiative populations for series 8
C           codes.  Modification of bgcoef.for
C
C  CALLING PROGRAM: Series 2 population codes.
C
C
C  PARAMETERS:
C           (I*4)  ndlev   = parameter = max. number of levels allowed
C           (I*4)  ndtrn   = parameter = max. no. of transitions allowed
C           (I*4)  ndtem   = parameter = max. no. of temperatures allowed
C           (I*4)  ndden   = parameter = max. number of densities allowed
C           (I*4)  ndmet   = parameter = max. no. of metastables allowed
C
C
C  INPUT :  (I*4)  nmet    = number of metastables (1 <= nmet <= 'ndmet')
C  INPUT :  (I*4)  imetr() = index of metastable in complete level list
C                            (array size = 'ndmet' )
C  INPUT :  (I*4)  nord    = number of ordinary levels (1 <= nmet <= 'ndmet')
C  INPUT :  (I*4)  iordr() = index of metastable in complete level list
C                            (array size = 'ndmet' )
C
C  INPUT :  (I*4)  maxt    = number of input temperatures ( 1 -> 'ndtem')
C  INPUT :  (R*8)  tine()  = electron temperatures (units: see 'ifout')
C  INPUT :  (R*8)  tinp()  = proton temperatures   (units: see 'ifout')
C  INPUT :  (R*8)  tinh()  = neutral hydrogen temperatures
C  INPUT :  (I*4)  ifout   = 1 => input temperatures in kelvin
```

```
C                           = 2 => input temperatures in ev
C                           = 3 => input temperatures in reduced form
C
C  INPUT :  (I*4)  maxd    = number of input densities ( 1 -> 'ndden')
C  INPUT :  (R*8)  dine()  = electron densities  (units: see 'idout')
C  INPUT :  (R*8)  dinp()  = proton densities    (units: see 'idout')
C  INPUT :  (I*4)  idout   = 1 => input densities in cm-3
C                           = 2 => input densities in reduced form
C
C  INPUT :  (L*4)  lpsel   = .true.  => include proton collisions
C                           = .false. =>do not include proton collisions
C  INPUT :  (L*4)  lzsel   = .true.  => scale proton collisions with
C                                       plasma z effective'zeff'.
C                           = .false. => do not scale proton collisions
C                                       with plasma z effective 'zeff'.
C                             (only used if 'lpsel=.true.')
C  INPUT :  (L*4)  liosel  = .true.  => include ionisation rates
C                           = .false. => do not include ionisation rates
C
C  INPUT :  (L*4)  lhsel   = .true.  => include charge transfer from
C                                       neutral hydrogren.
C                           = .false. => do not include charge transfer
C                                       from neutral hydrogren.
C  INPUT :  (L*4)  lrsel   = .true.  => include free electron
C                                       recombination.
C                           = .false. => do not include free electron
C                                       recombination.
C  INPUT :  (L*4)  lisel   = .true.  => include electron impact
C                                       ionisation.
C                           = .false. => do not include free electron
C                                       recombination.
C  INPUT :  (L*4)  lnsel   = .true.  => include projected bundle-n data
C                                       from datafile if available
C                           = .false. => do not include projected bundle-n
C                                       data
C  INPUT :  (I*4)  iz      = recombined ion charge
C  INPUT :  (I*4)  iz0     = nuclear charge
C  INPUT :  (I*4)  iz1     = recombining ion charge
C                             (note: iz1 should equal iz+1)
C
C  INPUT :  (I*4)  npl     = no. of metastables of (z+1) ion accessed
C                            by excited state ionisation in copase
C                            file with ionisation potentials given
C                            on the first data line
C  INPUT :  (R*8)  bwno    = ionisation potential (cm-1) of lowest parent
C  INPUT :  (R*8)  bwnoa() = ionisation potential (cm-1) of parents
C  INPUT :  (R*8)  prtwta()= weight for parent associated with bwnoa()
C
C  INPUT :  (I*4)  npla()  = no. of parent/zeta contributions to ionis.
C                            of level
C  INPUT :  (I*4)  ipla(,) = parent index for contributions to ionis.
C                            of level
C                            1st dimension: parent index
C                            2nd dimension: level index
C  INPUT :  (I*4)  zpla(,) = eff. zeta param. for contributions to ionis.
C                            of level
C                            1st dimension: parent index
C                            2nd dimension: level index
C  INPUT :  (I*4)  nplr    = no. of active metastables of (z+1) ion
C  INPUT :  (I*4)  npli    = no. of active metastables of (z-1) ion
C
```

```
C  INPUT :  (I*4)  il      = input data file: number of energy levels
C  INPUT :  (I*4)  maxlev  = highest index level in read transitions
C  INPUT :  (R*8)  xja()   = quantum number (j-value) for level 'ia()'
C                            note: (2*xja)+1 = statistical weight
C  INPUT :  (R*8)  wa()    = energy relative to level 1 (cm-1)
C                            dimension: level index
C  INPUT :  (R*8)  zeff    = plasma z effective ( if 'lzsel' = .true.)
C                            (if 'lzsel' = .false. => 'zeff=1.0')
C
C  INPUT :  (R*8)  xia()   = energy relative to ion. pot. (rydbergs)
C                            dimension: level index
C           (R*8)  er()    = energy relative to level 1 (rydbergs)
C                            dimension: level index
C
C  INPUT :  (I*4)  icnte   = number of electron impact transitions input
C  INPUT :  (I*4)  icntp   = number of proton impact transitions input
C  INPUT :  (I*4)  icntr   = number of free electron recombinations input
C  INPUT :  (I*4)  icnth   = no. of charge exchange recombinations input
C  INPUT :  (I*4)  icnti   = no. of ionisations to z input
C  INPUT :  (I*4)  icntl   = no. of satellite dr recombinations input
C  INPUT :  (I*4)  icnts   = no. of ionisations to z+1 input
C
C  INPUT :  (I*4)  ietrn() = electron impact transition:
C                            index values in main transition arrays which
C                            represent electron impact transitions.
C  INPUT :  (I*4)  iptrn() = proton impact transition:
C                            index values in main transition arrays which
C                            represent proton impact transitions.
C  INPUT :  (I*4)  irtrn() = free electron recombination:
C                            index values in main transition arrays which
C                            represent free electron recombinations.
C  INPUT :  (I*4)  ihtrn() = charge exchange recombination:
C                            index values in main transition arrays which
C                            represent charge exchange recombinations.
C  INPUT :  (I*4)  iitrn() = electron impact ionisation:
C                            index values in main transition arrays which
C                            represent ionisations from the lower stage
C  INPUT :  (I*4)  iltrn() = satellite dr recombination:
C                            index values in main transition arrays which
C                            represent satellite dr recombinations.
C  INPUT :  (I*4)  istrn() = electron impact ionisation:
C                            index values in main transition arrays which
C                            represent ionisations to upper stage ion.
C
C  INPUT :  (I*4)  ie1a()  = electron impact transition:
C                             lower energy level index
C  INPUT :  (I*4)  ie2a()  = electron impact transition:
C                             upper energy level index
C  INPUT :  (I*4)  ip1a()  = proton impact transition:
C                             lower energy level index
C  INPUT :  (I*4)  ip2a()  = proton impact transition:
C                             upper energy level index
C  INPUT :  (R*8)  aa()    = electron impact transition: a-value (sec-1)
C
C  INPUT :  (I*4)  ia1a()  = auger transition:
C                             parent energy level index
C  INPUT :  (I*4)  ia2a()  = auger transition:
C                             recombined ion energy level index
C  INPUT :  (R*8)  auga()  = auger transition: aug-value (sec-1)
C                             recombined ion energy level index
C  INPUT :  (I*4)  il1a()  = satellite dr transition:
```

```
C                               recomnining ion  index
C  INPUT :  (I*4)  il2a()  = satellite dr transition:
C                               recombined ion index
C  INPUT :  (R*8)  wvla()  = satellite dr transition: parent wvlgth.(a)
C                               dr satellite line index
C  INPUT :  (I*4)  is1a()  = ionising transition:
C                               ionised ion  index
C  INPUT :  (I*4)  is2a()  = ionising transition:
C                               ionising ion index
C  INPUT :  (L*4)  lss04a(,)= .true. => ionis. rate set in adf04 file:
C                               .false.=> not set in adf04 file
C                               1st dim: level index
C                               2nd dim: parent metastable index
C
C  INPUT :  (I*4)  nv      = input data file: number of gamma/temperature
C                              pairs for a given transition.
C  INPUT :  (R*8)  scef()  = input data file: electron temperatures (k)
C  INPUT :  (R*8)  scom(,) = transition:
C                               gamma values    input :  (case ' ' & 'p')
C                               rate coefft. (cm3 sec-1) (case 'h' & 'r')
C                              1st dimension - temperature 'scef()'
C                              2nd dimension - transition number
C
C  INPUT :  (C*44) dsnexp  = expansion data set name
C  INPUT :  (C*44) dsninc  = input copase data set name (mvs dsn)
C  INPUT :  (I*4)  iunt27  = output unit for results from expansion routine
C  INPUT :  (L*4)  open27  = .true.  => file allocated to unit 7.
C                          = .false. => no file allocated to unit 7.
C
C
C
C  OUTPUT : (R*8)
C
C
C
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          b8getp     ADAS      Fetch expansion data
C          bxchkm     ADAS      Checks if transition exist to metastable
C          bxiord     ADAS      Sets  up ordinary level index.
C          bxrate     ADAS      Calculates exc. & de-exc. rate coeffts.
C          b8loss     ADAS      Calculates direct line power loss
C          b8rcom     ADAS      Establishes recombination rate coeffts.
C          bxmcra     ADAS      Constructs a-value matrix.
C          bxmcrc     ADAS      Constructs exc./de-exc. rate coef matrix
C          b8mcca     ADAS      Constructs whole rate matrix.
C          bxmcma     ADAS      Constructs ordinary level rate matrix.
C          bxstka     ADAS      Stack up ordinary pop. dependence on met
C          b8stkb     ADAS      Stack up recomb. contribution for ord.
C          bxstkc     ADAS      Stack up transition rate between mets.
C          b8stkd     ADAS      Stack up recomb rate for each met. level
C          b8stke     ADAS      Stack up recomb(+3-body) contri.for ord.
C          b8stkf     ADAS      Stack up recomb(+3-body) for each met.
C          bxmpop     ADAS      Calculate basic met. level populations.
C          b8stvm     ADAS      Calculate met. level recomb. coeffts.
C          xxtcon     ADAS      Converts ispf entered temps. to ev.
C          xxdcon     ADAS      Converts ispf entered dens. to cm-3.
C          xxrate     ADAS      Calculates exc. & de-exc. rate coeffts.
C                               For unit gamma value.
```

```
C          xxminv     ADAS      Inverts matrix and solves equations.
C
C
C
C  This is a subroutine version of adas208 without the search for ionisation
C  rates from adf07 files.  Modified original 'bgcoef.for' subroutine by
C  Martin O'Mullane to add extra variables to the returned parameter set.
C
C  AUTHOR:  H. P. Summers, University of Strathclyde
C          JA8.08
C          Tel. 0141-553-4196
C
C  DATE:    20/04/02
C
C  UPDATE:
C
C  VERSION  : 1.1
C  DATE     : 20-01-2003
C  MODIFIED : Martin O'Mullane
C             - based on bgcoef v 1.1.
C             - Remove calculation of populations and lsseta as input.
C
C  VERSION  : 1.2
C  DATE     : 29-05-2003
C  MODIFIED : Martin O'Mullane
C             - Increase NDLEV to 1200.
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      CHARACTER*80        DSNEXP,      DSNINC
      INTEGER             I1A(NDTRN),  I2A(NDTRN),  IA1A(NDTRN)
      INTEGER             IA2A(NDTRN), ICNTE,       ICNTH,       ICNTI
      INTEGER             ICNTL,       ICNTP,       ICNTR,       ICNTS
      INTEGER             IDDEN,       IDLEV,       IDMET,       IDOUT
      INTEGER             IDTEM,       IDTRN,       IE1A(NDTRN)
      INTEGER             IE2A(NDTRN), IETRN(NDTRN),             IFOUT
      INTEGER             IHTRN(NDTRN),              IITRN(NDTRN)
      INTEGER             IL,          IL1A(NDLEV), IL2A(NDLEV)
      INTEGER             ILTRN(NDTRN),              IMETR(NDMET)
      INTEGER             IORDR(NDLEV),              IP1A(NDTRN)
      INTEGER             IP2A(NDTRN), IPLA(NDMET,NDLEV)
      INTEGER             IPTRN(NDTRN),              IRTRN(NDTRN)
      INTEGER             IS1A(NDLEV), IS2A(NDLEV), ISTRN(NDTRN)
      INTEGER             IUNT27,      IZ,          IZ0,         IZ1
      INTEGER             MAXD,        MAXLEV,      MAXT,        NMET
      INTEGER             NORD,        NPL,         NPLA(NDLEV), NPLI
      INTEGER             NPLR,        NV
      LOGICAL             LHSEL,       LIOSEL,      LISEL,       LNSEL
      LOGICAL             LPSEL,       LRSEL,       LSS04A(IDLEV,IDMET)
      LOGICAL             LTICK,       LZSEL,       OPEN27
      REAL*8              AA(NDTRN),   AUGA(NDTRN), BWNO
      REAL*8              BWNOA(NDMET),              DINE(NDDEN)
      REAL*8              DINP(NDDEN), ER(NDLEV),   PRTWTA(NDMET)
      REAL*8              SCEF(14),    SCOM(14,NDTRN)
      REAL                STACK(IDLEV,IDMET,IDTEM,IDDEN)
      REAL*8              STCKM(IDMET,IDTEM,IDDEN)
      REAL                STVH(IDLEV,IDTEM,IDDEN,IDMET)
      REAL*8              STVHM(IDMET,IDTEM,IDDEN,IDMET)
      REAL                STVI(IDLEV,IDTEM,IDDEN,IDMET)
      REAL*8              STVIM(IDMET,IDTEM,IDDEN,IDMET)
      REAL                STVR(IDLEV,IDTEM,IDDEN,IDMET)
```

```
REAL*8              STVRM(IDMET,IDTEM,IDDEN,IDMET)
REAL*8              TINE(NDTEM), TINH(NDTEM), TINP(NDTEM)
REAL*8              WA(NDLEV),   WVLA(NDLEV), XIA(NDLEV)
REAL*8              XJA(NDLEV),  ZEFF,        ZPLA(NDMET,NDLEV)
```

3.102 bxcstr: Subroutine bxcstr from library adas2xx

```
       SUBROUTINE BXCSTR( CSTRGA , IL     , IMAX  ,
     &                    CSTRGB
     &                  )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: BXCSTR *********************
C
C  PURPOSE:  TO TAKE THE LAST 'IMAX' NON-BLANK BYTES OF THE 'CSTRGA'
C            STRING AND PLACE THEM IN THE 'CSTRGB' STRING.
C
C  CALLING PROGRAM: ADAS205/ADAS206
C
C
C  SUBROUTINE:
C
C  INPUT : (C*(*))CSTRGA()= NOMENCLATURE/CONFIGURATION FOR LEVEL 'IA()'
C  INPUT : (I*4)  IL      = NUMBER OF ENERGY LEVELS
C  INPUT : (I*4)  IMAX    = NUMBER OF NON-BLANK BYTES TO BE EXTRACTED
C                           FROM THE END OF THE INPUT STRING.
C
C  OUTPUT: (C*(*))CSTRGB()= LAST 'IMAX' NON-BLANK BYTES FROM 'CSTRGA()'
C
C          (I*4)  IFIRST  = POSITION OF FIRST NON-BLANK BYTE IN CSTRGA()
C          (I*4)  ILAST   = POSITION OF LAST  NON-BLANK BYTE IN CSTRGA()
C          (I*4)  ILEN    = LENGTH IN BYTES OF NON-BLANK PART OF CSTRGA
C          (I*4)  I       = GENERAL USE (ARRAY INDEX)
C
C
C ROUTINES:
C           ROUTINE     SOURCE      BRIEF DESCRIPTION
C           ------------------------------------------------------------
C           XXSLEN      ADAS        FIRST/LAST NONBLANK BYTES IN STRING
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/37
C          JET EXT. 5023
C
C DATE:    06/07/93
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      CHARACTER*(*)       CSTRGA(IL),  CSTRGB(IL)
      INTEGER             IL,          IMAX
```

3.103 bxdata: Subroutine bxdata from library adas2xx

```
       SUBROUTINE BXDATA( IUNIT  , NDLEV  , NDTRN ,
     &                    TITLED , IZ     , IZ0   , IZ1   , BWNO  ,
     &                    IL     ,
     &                    IA     , CSTRGA , ISA   , ILA   , XJA   , WA ,
     &                    NV     , SCEF   ,
     &                    ITRAN  , MAXLEV ,
     &                    TCODE  , I1A    , I2A   , AVAL  , SCOM
     &                  )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: BXDATA *********************
C
C  PURPOSE:  TO FETCH DATA FROM INPUT COPASE DATA SET.
C
C  CALLING PROGRAM: ADAS205/ADAS206/ADAS201
C
C  DATA:
C           THE 'REAL' DATA IN THE FILE IS REPRESENTED IN AN ABBREVIATED
C           FORM WHICH OMITS THE "D" OR "E" EXPONENT SPECIFIER.
C           e.g. 1.23D-06 or 1.23E-06 IS REPRESENTED AS 1.23-06
C                6.75D+07 or 6.75E+07 IS REPRESENTED AS 6.75+07
C
C           THEREFORE THE FORM OF EACH 'REAL' NUMBER IN THE DATA SET IS:
C                          N.NN+NN or N.NN-NN
C
C           THE UNITS USED IN THE DATA FILE ARE TAKEN AS FOLLOWS:
C
C           IONISATION POTENTIAL: WAVE NUMBER (CM-1)
C           INDEX LEVEL ENERGIES: WAVE NUMBER (CM-1)
C           TEMPERATURES        : KELVIN
C           A-VALUES            : SEC-1
C           GAMMA-VALUES        :
C           RATE COEFFT.        : CM3 SEC-1
C
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  IUNIT   = UNIT TO WHICH INPUT FILE IS ALLOCATED
C  INPUT : (I*4)  NDLEV   = MAXIMUM NUMBER OF LEVELS THAT CAN BE READ
C  INPUT : (I*4)  NDTRN   = MAX. NUMBER OF TRANSITIONS THAT CAN BE READ
C
C  OUTPUT: (C*3)  TITLED  = ELEMENT SYMBOL.
C  OUTPUT: (I*4)  IZ      =  RECOMBINED ION CHARGE READ
C  OUTPUT: (I*4)  IZ0     =         NUCLEAR CHARGE READ
C  OUTPUT: (I*4)  IZ1     = RECOMBINING ION CHARGE READ
C                           (NOTE: IZ1 SHOULD EQUAL IZ+1)
C  OUTPUT: (R*8)  BWNO    = IONISATION POTENTIAL (CM-1)
C
C  OUTPUT: (I*4)  IL      = INPUT DATA FILE: NUMBER OF ENERGY LEVELS
C
C  OUTPUT: (I*4)  IA()    = ENERGY LEVEL INDEX NUMBER
C  OUTPUT: (C*18) CSTRGA()= NOMENCLATURE/CONFIGURATION FOR LEVEL 'IA()'
C  OUTPUT: (I*4)  ISA()   = MULTIPLICITY FOR LEVEL 'IA()'
C                           NOTE: (ISA-1)/2 = QUANTUM NUMBER (S)
C  OUTPUT: (I*4)  ILA()   = QUANTUM NUMBER (L) FOR LEVEL 'IA()'
C  OUTPUT: (R*8)  XJA()   = QUANTUM NUMBER (J-VALUE) FOR LEVEL 'IA()'
C                           NOTE: (2*XJA)+1 = STATISTICAL WEIGHT
C  OUTPUT: (R*8)  WA()    = ENERGY RELATIVE TO LEVEL 1 (CM-1) FOR LEVEL
C                           'IA()'
```

```
C
C  OUTPUT: (I*4)  NV      = INPUT DATA FILE: NUMBER OF GAMMA/TEMPERATURE
C                            PAIRS FOR A GIVEN TRANSITION.
C  OUTPUT: (R*8)  SCEF()  = INPUT DATA FILE: ELECTRON TEMPERATURES (K)
C                            (INITIALLY JUST THE MANTISSA. SEE 'ITPOW()')
C                            (NOTE: TE=TP=TH IS ASSUMED)
C
C  OUTPUT: (I*4)  ITRAN   = INPUT DATA FILE: NUMBER OF TRANSITIONS
C  OUTPUT: (I*4)  MAXLEV  = HIGHEST INDEX LEVEL IN READ TRANSITIONS
C
C  OUTPUT: (C*1)  TCODE() = TRANSITION: DATA TYPE POINTER:
C                            ' ' => Electron Impact   Transition
C                            'P' => Proton   Impact   Transition
C                            'H' => Charge   Exchange Recombination
C                            'R' => Free     Electron Recombination
C  OUTPUT: (I*4)  I1A()   = TRANSITION:
C                             LOWER ENERGY LEVEL INDEX (CASE ' ' & 'P')
C                             SIGNED PARENT      INDEX (CASE 'H' & 'R')
C  OUTPUT: (I*4)  I2A()   = TRANSITION:
C                             UPPER ENERGY LEVEL INDEX (CASE ' ' & 'P')
C                             CAPTURING    LEVEL INDEX (CASE 'H' & 'R')
C  OUTPUT: (R*8)  AVAL()  = TRANSITION:
C                             A-VALUE (SEC-1)          (CASE ' ')
C                             NEUTRAL BEAM ENERGY      (CASE 'H')
C                             NOT USED                 (CASE 'P' & 'R')
C  OUTPUT: (R*8)  SCOM(,) = TRANSITION:
C                             GAMMA VALUES             (CASE ' ' & 'P')
C                             RATE COEFFT. (CM3 SEC-1) (CASE 'H' & 'R')
C                            1ST DIMENSION - TEMPERATURE 'SCEF()'
C                            2ND DIMENSION - TRANSITION NUMBER
C
C          (I*4)  NVMAX   = PARAMETER = MAX. NUMBER OF TEMPERATURES
C                                       THAT CAN BE READ IN.
C          (I*4)  MTIED   = PARAMETER = MUST BE GREATER THAN OR EQUAL TO
C                                       THE MAX. NO. OF LEVELS.
C          (R*8)  DZERO   = PARAMETER = MINIMUM VALUE FOR 'AVAL()' AND
C                                       'SCOM()' ARRAYS = 1.0D-30
C
C          (I*4)  I4UNIT  = FUNCTION (SEE ROUTINE SECTION BELOW)
C          (I*4)  IQS     = X-SECT DATA FORMAT SELECTOR
C                           NOTE: IQS=3 ONLY ALLOWED IN THIS PROGRAM
C          (I*4)  I       = GENERAL USE.
C          (I*4)  IABT    = RETURN CODE FROM 'R8FCTN' (0 => NO ERROR)
C                           OR FROM INTERROGATION OF 'C7'
C          (I*4)  IFIRST  = BYTE POSITION OF START OF NUMBER IN BUFFER
C          (I*4)  ILAST   = BYTE POSITION OF END   OF NUMBER IN BUFFER
C          (I*4)  IWORD   = THE WORD POSITION OF THE REQUIRED DATA IN
C                           A STRING TO BE INTERROGATED BY XXWORD.
C          (I*4)  J       = GENERAL USE.
C          (I*4)  J1      = INPUT DATA FILE - SELECTED TRANSITION:
C                             LOWER ENERGY LEVEL INDEX (CASE ' ' & 'P')
C          (I*4)  J2      = INPUT DATA FILE - SELECTED TRANSITION:
C                             UPPER ENERGY LEVEL INDEX (CASE ' ' & 'P')
C                             CAPTURING    LEVEL INDEX (CASE 'H' & 'R')
C          (I*4)  LENCST  = BYTE LENGTH OF STRING CSTRGA()
C          (I*4)  NWORDS  = NUMBER OF NUMBERS STORED IN BUFFER
C          (I*4)  ILINE   = ENERGY LEVEL INDEX FOR CURRENT LINE
C          (I*4)  IAPOW   = EXPONENT OF 'AVALM'
C          (I*4)  IGPOW() = EXPONENT OF 'GAMMA()'
C          (I*4)  ITPOW() = TEMPERATURES - EXPONENT
C                           NOTE: MANTISSA INITIALLY KEPT IN 'SCEF()'
```

```
C
C          (R*4)  ZF      = SHOULD BE EQUIVALENT TO 'IZ1'
C
C          (R*8)  AVALM   = INPUT DATA FILE - SELECTED TRANSITION:
C                           MANTISSA OF:   ('IAPOW' => EXPONENT)
C                            A-VALUE (SEC-1)          (CASE ' ')
C                            NEUTRAL BEAM ENERGY      (CASE 'H')
C                            NOT USED                 (CASE 'P' & 'R')
C          (R*8)  GAMMA() = INPUT DATA FILE - SELECTED TRANSITION:
C                           MANTISSA OF: ('IGPOW()' => EXPONENT)
C                            GAMMA VALUES             (CASE ' ' & 'P')
C                            RATE COEFFT. (CM3 SEC-1) (CASE 'H' & 'R')
C                           DIMENSION => TEMPERATURE 'SCEF()'
C
C          (C*7)  C7      = USED TO PARSE VALUE FOR XJA()
C          (C*7)  CDELIM  = DELIMITERS FOR INPUT OF DATA FROM HEADERS
C          (C*18) C18     = USED TO PARSE VALUE TO CSTRGA()
C          (C*80) CLINE   = CURRENT ENERGY LEVEL INDEX PARAMETER LINE
C          (C*128)BUFFER  = GENERAL STRING BUFFER STORAGE
C          (C*3)  CITPOW()= USED TO PARSE VALUES TO ITPOW()
C          (C*5)  CSCEF() = USED TO PARSE VALUES TO SCEF()
C          (C*7)  CFORM7  = FORMAT FOR INTERNAL READING OF REAL NUMBER
C
C          (L*4)  LDATA   = IDENTIFIES  WHETHER  THE END OF AN  INPUT
C                           SECTION IN THE DATA SET HAS BEEN LOCATED.
C                           (.TRUE. => END OF SECTION REACHED)
C          (L*4)  LTCHR   = .TRUE.  => CURRENT 'TCODE()' = 'H' OR 'R'.
C                         = .FALSE. => CURRENT 'TCODE()'.NE.'H' OR 'R'.
C          (L*4)  LTCPR   = .TRUE.  => CURRENT 'TCODE()' = 'P' OR 'R'.
C                         = .FALSE. => CURRENT 'TCODE()'.NE.'P' OR 'R'.
C          (L*4)  LERROR  = .TRUE.  => UNTIED LEVEL FOUND
C                         = .FALSE. => ALL LEVELS TIED
C          (L*4)  LTIED() = .TRUE.  => SPECIFIED LEVEL TIED
C                         = .FALSE. => SPECIFIED LEVEL IS UNTIED
C                           DIMENSION => LEVEL INDEX
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          XXWORD     ADAS      EXTRACT POSITION OF NUMBER IN BUFFER
C          I4UNIT     ADAS      FETCH UNIT NUMBER FOR OUTPUT OF MESSAGES
C          R8FCTN     ADAS      CONVERTS FROM CHARACTER TO REAL VARIABLE
C
C NOTE:             LTCHR        LTCPR         TCODE()
C                  ------------------------------------
C                  .TRUE.       .TRUE.    =>     'R'
C                  .TRUE.       .FALSE.   =>     'H'
C                  .FALSE.      .TRUE.    =>     'P'
C                  .FALSE.      .FALSE.   =>     ' '
C
C        FOR A-VALUES & GAMMA-VALUES ENTRIES LESS THAN 'DZERO' ARE TAKEN
C        AS BEING EQUAL TO DZERO. THIS AFFECTS THE 'AVAL()' AND 'SCOM()'
C        ARRAYS.
C
C ROUTINES: NONE
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/37
C          JET EXT. 5023
C
C DATE:    09/10/90
```

```
C
C UPDATE:  16/11/90 - LEVEL LINE READ AS A CHARACTER*80 STRING FIRST
C                     (PE BRIDEN)
C
C UPDATE:  01/05/92 - CHECK MADE TO MAKE SURE NO UNTIED LEVEL EXISTS.
C                     IF UNTIED LEVELS EXIST PROGRAM IS TERMINATED
C                     WITH A MESSAGE.
C                     (PE BRIDEN)
C
C UPDATE:  26/06/92 - INCREASED PARAMETER MTIED FROM 100 TO 200
C
C UPDATE:  30/07/92 - INPUT VARIABLE 'XJA' NOW ALLOWED TO HAVE A LENGTH
C                     OF BETWEEN 1 AND 6 STARTING AT COLUMN 30 - IT MUST
C                     BE FOLLOWED BY A ')' WHICH CANNOT BE PLACED AFTER
C                     COLUMN 36. INTRODUCED VARIABLE 'C7' TO PARSE VALUE
C                     AND USE FUNCTION R8FCTN TO INTERROGATE C7.
C                     - EDITED FORMAT STATEMENT 1003 ACCORDINGLY.
C                     - INTRODUCED FORMAT STATEMENT 1012.
C
C UPDATE:  23/04/93 - PE BRIDEN - ADAS91: ADDED I4UNIT FUNCTION TO WRITE
C                                         STATEMENTS FOR SCREEN MESSAGES
C
C UPDATE:  24/05/93 - PE BRIDEN - ADAS91: CHANGED I4UNIT(0)-> I4UNIT(-1)
C
C UPDATE:  20/05/93 - PE BRIDEN - ADAS91: MAJOR REVISION -
C                                         MODIFIED TO READ IN NEW INPUT
C                                         DATA-SET STYLE AND ALSO ALLOW
C                                         THE OLD-STYLE TO BE READ.
C                                         ARGUMENT DIMENSIONS CHANGED
C                                         BUT CODE ROUTINE SHOULD SPOT
C                                         CASES WHERE THE ORIGINAL
C                                         ARGUMENT DIMS ARE USED AND
C                                         ACT ACCORDINGLY.
C
C UPDATE:  05/08/93 - PE BRIDEN - ADAS91: MINOR REVISION -
C                                         IF DATA TYPE POINTER (TCODE())
C                                         EQUALS 'H' (Charge Exchange
C                                         Recomb.) or 'R' (Free Electron
C                                         Recomb.) - I1A()  now stores
C                                         the signed parent index(see
C                                         I1A() above)
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 04-05-93
C MODIFIED: ANDREW BOWEN (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST VERSION
C
C VERSION: 1.2                          DATE: 04-05-93
C MODIFIED: ANDREW BOWEN
C               - ERROR WRIES CHANGED TO UNIT 0.
C
C VERSION: 1.3                          DATE: 21-03-95
C MODIFIED: LALIT JALOTA (TESSELLA SUPPORT SERVICES PLC)
C               -
C
C VERSION: 1.4                          DATE: 23-03-95
C MODIFIED: LALIT JALOTA
C               -
C
C VERSION: 1.5                          DATE: 02-04-96
C MODIFIED: TIM HAMMOND/PAUL BRIDEN (TESSELLA SUPPORT SERVICES PLC)
```

```
C               - INSTEAD OF USING FORMAT SPECIFIER F15.0 WHEN
C                 INTERNALLY READING A FLOATING POINT NUMBER,
C                 CREATE THE APPROPRIATE SPECIFIER WITHIN CFORM7
C                 AND USE THIS.
C
C VERSION: 1.6 DATE: 24-06-97
C MODIFIED: HUGH SUMMERS
C               - CHANGED PARAMETER MTIED FROM 200 TO 300
C
C VERSION: 1.7 DATE: 26-02-97
C MODIFIED: M.O'MULLANE AND R. MARTIN
C     - CHANGED 'I2' TO 'I4' TO IN FORMAT STATEMENT 1011
C
C VERSION: 1.8 DATE: 20-09-99
C MODIFIED: R. MARTIN
C     - CHANGED 'I3' TO 'I4' TO IN FORMAT STATEMENT 1001
C
C VERSION: 1.9 DATE: 28-05-2003
C MODIFIED: Martin O'Mullane
C     - Warn user that the routine is now deprecated
C                     and that xxdata_04 should be used instead.
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      CHARACTER*(*)       CSTRGA(NDLEV)
      CHARACTER           TCODE(NDTRN)
      CHARACTER*3         TITLED
      INTEGER             I1A(NDTRN),  I2A(NDTRN),  IA(NDLEV),   IL
      INTEGER             ILA(NDLEV),  ISA(NDLEV),  ITRAN,       IUNIT
      INTEGER             IZ,          IZ0,         IZ1,         MAXLEV
      INTEGER             NDLEV,       NDTRN,       NV
      REAL*8              AVAL(NDTRN), BWNO,        SCEF(NVMAX)
      REAL*8              SCOM(NVMAX,NDTRN),        WA(NDLEV)
      REAL*8              XJA(NDLEV)
```

3.104 bxiord: Subroutine bxiord from library adas2xx

```
      SUBROUTINE BXIORD( IL   ,
     &                   NMET , IMETR ,
     &                   NORD , IORDR
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: BXIORD *********************
C
C  PURPOSE: TO SET UP THE INDEX OF ORDINARY EXCITED LEVELS IN COMPLETE
C           LEVEL LIST 'IORDR()'.
C
C  CALLING PROGRAM:  ADAS205/ADAS206
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  IL      = NUMBER OF ENERGY LEVELS (MET. & ORD.)
C
C  INPUT : (I*4)  NMET    = NUMBER OF METASTABLE STATES
C  INPUT : (I*4)  IMETR() = INDEX OF METASTABLES IN COMPLETE LEVEL LIST
C
C  OUTPUT: (I*4)  NORD    = NUMBER OF ORDINARY EXCITED LEVELS.
C  OUTPUT: (I*4)  IORDR() = INDEX OF ORDINARY EXCITED LEVELS IN COMPLETE
C                           LEVEL LIST.
C
C          (I*4)  I4UNIT  = FUNCTION (SEE ROUTINE SECTION BELOW)
C          (I*4)  IS      = ENERGY LEVEL ARRAY INDEX
C          (I*4)  IM      = METASTABLE LEVEL NUMBER COUNTER
C          (I*4)  IO      = ORDINARY EXCITED LEVEL NUMBER COUNTER
C
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          XXTERM     ADAS      TERMINATES PROGRAM WITH MESSAGE
C          I4UNIT     ADAS      FETCH UNIT NUMBER FOR OUTPUT OF MESSAGES
C
C
C NOTE:    'NMET' + 'NORD' = 'IL'
C
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 4569
C
C DATE:    09/10/90
C
C UPDATE:  23/04/93 - PE BRIDEN - ADAS91: ADDED I4UNIT FUNCTION TO WRITE
C                                         STATEMENTS FOR SCREEN MESSAGES
C
C UPDATE:  24/05/93 - PE BRIDEN - ADAS91: CHANGED I4UNIT(0)-> I4UNIT(-1)
C
C VERSION: 1.5 DATE: 26-06-97
C MODIFIED: H.P. SUMMERS, RICHARD MARTIN
C - CHANGED LINE IM=IM+1 TO IM=MIN(IM+1,NMET-1)
C   THIS ENSURES UPPER ARRAY BOUND OF IMETR IS NOT
C   EXCEEDED.
C
C-----------------------------------------------------------------------
C
```

```
C-----------------------------------------------------------------------
      INTEGER             IL,          IMETR(NMET), IORDR(IL),   NMET
      INTEGER             NORD
```

3.105 bxmcca: Subroutine bxmcca from library adas2xx

```
      SUBROUTINE BXMCCA( NDLEV  , IL     ,
     &                            LPSEL  , LISEL ,
     &                   DENE   , DENP   ,
     &                   CRA    ,
     &                   CRCE   , CRCP   , CIE   ,
     &                   CC
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: BXMCCA *********************
C
C  PURPOSE: TO  CONSTRUCT WHOLE RATE MATRIX 'CC' FOR TRANSITIONS BETWEEN
C           ALL ENERGY LEVELS AT A FIXED TEMPERATURE  AND  GIVEN DENSITY
C           'DENE/DENP'.
C
C  CALLING PROGRAM:  ADAS205/ADAS206
C
C  SUBROUTINE:
C
C  INPUT :  (I*4)  NDLEV   = MAXIMUM NUMBER OF ENERGY LEVELS ALLOWED
C  INPUT :  (I*4)  IL      = NUMBER OF ENERGY LEVELS
C
C  INPUT :  (L*4)  LPSEL   = .TRUE.  => INCLUDE PROTON COLLISIONS
C                            .FALSE. => DO NOT INCLUDE PROTON COLLISIONS
C  INPUT :  (L*4)  LISEL   = .TRUE.  => INCLUDE IONISATION RATES
C                            .FALSE. => DO NOT INCLUDE IONISATION RATES
C
C  INPUT :  (R*8)  DENE    = ELECTRON DENSITY (UNITS: CM-3)
C  INPUT :  (R*8)  DENP    = PROTON DENSITY (UNITS: CM-3)
C
C  INPUT :  (R*8)  CRA(,)  = A-VALUE (sec-1)  MATRIX  COVERING   ALL
C                            TRANSITIONS.
C                            1st DIMENSION: ENERGY LEVEL INDEX
C                            2nd DIMENSION: ENERGY LEVEL INDEX
C                           (NOTE: DIAGONAL   ELEMENTS  REPRESENT   THE
C                                  NEGATIVE  SUM  OF  THEIR  RESPECTIVE
C                                  COLUMNS.)
C
C  INPUT :  (R*8)  CRCE(,) = ELECTRON IMPACT TRANSITIONS:
C                            EXCIT'N/DE-EXCIT'N RATE COEFFT MATRIX
C                            COVERING ALL TRANSITIONS (cm**3/s).
C                            VALUES FOR GIVEN TEMPERATURE.
C                            1st DIMENSION: ENERGY LEVEL INDEX
C                            2nd DIMENSION: ENERGY LEVEL INDEX
C                           (NOTE: DIAGONAL   ELEMENTS  REPRESENT   THE
C                                  NEGATIVE  SUM  OF  THEIR  RESPECTIVE
C                                  COLUMNS.)
C  INPUT :  (R*8)  CRCP(,) = PROTON IMPACT TRANSITIONS:
C                            EXCIT'N/DE-EXCIT'N RATE COEFFT MATRIX
C                            COVERING ALL TRANSITIONS (cm**3/s).
C                            VALUES FOR GIVEN TEMPERATURE.
C                            1st DIMENSION: ENERGY LEVEL INDEX
C                            2nd DIMENSION: ENERGY LEVEL INDEX
C                           (NOTE: DIAGONAL   ELEMENTS  REPRESENT   THE
C                                  NEGATIVE  SUM  OF  THEIR  RESPECTIVE
C                                  COLUMNS.)
C  INPUT :  (R*8)  CIE()   = IONISATION RATE COEFFICIENT VECTOR FOR
C                            FIXED TEMPERATURE.
C                            DIMENSION: ENERGY LEVEL INDEX
```

```
C
C  OUTPUT:  (R*8)  CC(,)    = RATE MATRIX COVERING ALL TRANSITIONS
C                             (UNITS: SEC-1)
C                             VALUES FOR GIVEN TEMPERATURE AND DENSITY.
C                             1st DIMENSION: ENERGY LEVEL INDEX
C                             2nd DIMENSION: ENERGY LEVEL INDEX
C
C           (I*4)  IS1      = ENERGY LEVEL ARRAY INDEX
C           (I*4)  IS2      = ENERGY LEVEL ARRAY INDEX
C
C
C ROUTINES: NONE
C
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 4569
C
C DATE:    09/10/90
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IL,          NDLEV
      LOGICAL             LISEL,       LPSEL
      REAL*8              CC(NDLEV,NDLEV),          CIE(NDLEV)
      REAL*8              CRA(NDLEV,NDLEV),         CRCE(NDLEV,NDLEV)
      REAL*8              CRCP(NDLEV,NDLEV),        DENE,        DENP
```

3.106 bxmcma: Subroutine bxmcma from library adas2xx

```
      SUBROUTINE BXMCMA( NDLEV  ,
     &                   NORD   , IORDR  ,
     &                   CC     ,
     &                   CMAT
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: BXMCMA *********************
C
C  PURPOSE: TO  STACK  UP  NON-METASTABLE/ORDINARY  EXCITED  LEVEL  RATE
C           MATRIX 'CMAT' FROM WHOLE RATE MATRIX 'CC' FOR ALL TRANSIT'NS
C           BETWEEN ALL ENERGY LEVELS AT A FIXED TEMPERATURE AND DENSITY
C
C  CALLING PROGRAM:  ADAS205/ADAS206
C
C  SUBROUTINE:
C
C  INPUT :  (I*4)  NDLEV   = MAXIMUM NUMBER OF ENERGY LEVELS ALLOWED
C
C  INPUT :  (I*4)  NORD    = NUMBER OF NON-METASTABLE/ORDINARY EXCITED
C                            ENERGY LEVELS.
C  INPUT :  (I*4)  IORDR() = INDEX OF NON-METASTABLE/ORDINARY EXCITED
C                            LEVELS IN COMPLETE LEVEL LIST.
C
C  INPUT :  (R*8)  CC(,)   = RATE MATRIX COVERING ALL TRANSITIONS
C                            (UNITS: SEC-1)
C                            VALUES FOR GIVEN TEMPERATURE AND DENSITY.
C                            1st DIMENSION: ENERGY LEVEL INDEX
C                            2nd DIMENSION: ENERGY LEVEL INDEX
C
C  OUTPUT:  (R*8)  CMAT(,) = RATE MATRIX COVERING ALL NON-METASTABLE/
C                            ORDINARY EXCITED LEVELS.
C                            (UNITS: SEC-1)
C                            VALUES FOR GIVEN TEMPERATURE AND DENSITY.
C                            1st DIMENSION: ORDINARY EXCITED LEVEL INDEX
C                            2nd DIMENSION: ORDINARY EXCITED LEVEL INDEX
C
C           (I*4)  IS1     = ORDINARY EXCITED LEVEL ARRAY INDEX
C           (I*4)  IS2     = ORDINARY EXCITED LEVEL ARRAY INDEX
C
C
C ROUTINES: NONE
C
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 4569
C
C DATE:    09/10/90
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IORDR(NDLEV),             NDLEV,       NORD
      REAL*8              CC(NDLEV,NDLEV),          CMAT(NDLEV,NDLEV)
```

3.107 bxmcra: Subroutine bxmcra from library adas2xx

```
      SUBROUTINE BXMCRA( NDTRN  , NDLEV  ,
     &                   ICNT   , IL     ,
     &                   I1A    , I2A    ,
     &                   AVAL   ,
     &                   CRA
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: BXMCRA *********************
C
C  PURPOSE: TO CONSTRUCT A-VALUE MATRIX  'CRA'  FOR TRANSITIONS  BETWEEN
C           ALL ENERGY LEVELS.
C
C  CALLING PROGRAM:  ASAS205/ADAS206
C
C  SUBROUTINE:
C
C  INPUT :  (I*4)  NDTRN   = MAXIMUM NUMBER OF RECOMBINATIONS ALLOWED
C  INPUT :  (I*4)  NDLEV   = MAXIMUM NUMBER OF ENERGY LEVELS ALLOWED
C
C  INPUT :  (I*4)  ICNT    = NUMBER OF SELECTED TRANSITIONS
C  INPUT :  (I*4)  IL      = NUMBER OF ENERGY LEVELS
C                                        (SEE: 'ITRN()')
C
C  INPUT :  (I*4)  I1A()   = SELECTED TRANSITION TYPE:
C                            LOWER ENERGY LEVEL INDEX.
C                            DIMENSION: TRANSITION INDEX
C  INPUT :  (I*4)  I2A()   = SELECTED TRANSITION TYPE:
C                            UPPER ENERGY LEVEL INDEX.
C                            DIMENSION: TRANSITION INDEX
C
C  INPUT :  (R*8)  AVAL()  = A-VALUE (sec-1)
C                            DIMENSION: TRANSITION INDEX
C
C  OUTPUT:  (R*8)  CRA(,)  = A-VALUE  (sec-1)  MATRIX  COVERING  ALL
C                            TRANSITIONS.
C                            1st DIMENSION: LOWER ENERGY LEVEL INDEX
C                            2nd DIMENSION: UPPER ENERGY LEVEL INDEX
C                           (NOTE: DIAGONAL   ELEMENTS  REPRESENT   THE
C                                  NEGATIVE  SUM  OF  THEIR  RESPECTIVE
C                                  COLUMNS.)
C
C           (I*4)  IS1     = ENERGY LEVEL ARRAY INDEX
C           (I*4)  IS2     = ENERGY LEVEL ARRAY INDEX
C           (I*4)  IC      = TRANSITION ARRAY INDEX
C
C ROUTINES: NONE
C
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 4569
C
C DATE:    09/10/90
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             I1A(NDTRN),  I2A(NDTRN),  ICNT,        IL
```

```
INTEGER             NDLEV,       NDTRN
REAL*8              AVAL(NDTRN), CRA(NDLEV,NDLEV)
```

3.108 bxmcrc: Subroutine bxmcrc from library adas2xx

```
      SUBROUTINE BXMCRC( NDTEM  , NDTRN  , NDLEV  ,
     &                   IT     , ICNT   , IL     ,
     &                   I1A    , I2A    ,
     &                   RATE   , DRATE  ,
     &                   CRC
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: BXMCRC *********************
C
C  PURPOSE: TO  CONSTRUCT  EXCITATION/DE-EXCIATATION   RATE  COEFFICIENT
C           MATRIX  'CRC' FOR TRANSITIONS BETWEEN ALL ENERGY LEVELS AT A
C           GIVEN TEMPERATURE 'IT' AND FOR A GIVEN TRANSITION TYPE
C
C  CALLING PROGRAM:  ADAS205/ADAS206
C
C  SUBROUTINE:
C
C  INPUT :  (I*4)  NDTEM   = MAXIMUM NUMBER OF TEMPERATURES ALLOWED
C  INPUT :  (I*4)  NDTRN   = MAXIMUM NUMBER OF RECOMBINATIONS ALLOWED
C  INPUT :  (I*4)  NDLEV   = MAXIMUM NUMBER OF ENERGY LEVELS ALLOWED
C
C  INPUT :  (I*4)  IT      = INDEX OF TEMPERATURE VALUE BEING ASSESSED
C  INPUT :  (I*4)  ICNT    = NUMBER OF SELECTED TRANSITIONS
C  INPUT :  (I*4)  IL      = NUMBER OF ENERGY LEVELS
C                                        (SEE: 'ITRN()')
C
C  INPUT :  (I*4)  I1A()   = SELECTED TRANSITION TYPE:
C                            LOWER ENERGY LEVEL INDEX.
C                            DIMENSION: TRANSITION INDEX
C  INPUT :  (I*4)  I2A()   = SELECTED TRANSITION TYPE:
C                            UPPER ENERGY LEVEL INDEX.
C                            DIMENSION: TRANSITION INDEX
C
C  INPUT :  (R*8)  RATE(,) = EXCITATION RATE COEFFS (cm**3/s)
C                            1st DIMENSION: TEMPERATURE INDEX
C                            2nd DIMENSION: TRANSITION INDEX
C  INPUT :  (R*8)  DRATE(,)= DE-EXCIT'N RATE COEFFS (cm**3/s)
C                            1st DIMENSION: TEMPERATURE INDEX
C                            2nd DIMENSION: TRANSITION INDEX
C
C  OUTPUT:  (R*8)  CRC(,)  = EXCIT'N/DE-EXCIT'N RATE COEFFT MATRIX
C                            COVERING ALL TRANSITIONS (cm**3/s).
C                            VALUES FOR GIVEN TEMPERATURE & TRANSITION
C                            TYPE.
C                            1st DIMENSION: ENERGY LEVEL INDEX
C                            2nd DIMENSION: ENERGY LEVEL INDEX
C                           (NOTE: DIAGONAL   ELEMENTS  REPRESENT   THE
C                                  NEGATIVE  SUM  OF  THEIR  RESPECTIVE
C                                  COLUMNS.)
C
C           (I*4)  IS1     = ENERGY LEVEL ARRAY INDEX
C           (I*4)  IS2     = ENERGY LEVEL ARRAY INDEX
C           (I*4)  IC      = TRANSITION ARRAY INDEX
C
C
C ROUTINES: NONE
C
C
```

```
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 4569
C
C DATE:    09/10/90
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             I1A(NDTRN),  I2A(NDTRN),  ICNT,         IL
      INTEGER             IT,          NDLEV,       NDTEM,        NDTRN
      REAL*8              CRC(NDLEV,NDLEV),         DRATE(NDTEM,NDTRN)
      REAL*8              RATE(NDTEM,NDTRN)
```

3.109 bxmpop: Subroutine bxmpop from library adas2xx

```
      SUBROUTINE BXMPOP( NDMET ,
     &                   NMET  ,
     &                   CRED  ,
     &                   RHS   , CRMAT ,
     &                   STKM
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: BXMPOP *********************
C
C  PURPOSE: TO CALCULATE AND STACK UP IN  'STKM'  THE  METASTABLE LEVEL
C           POPULATIONS FOR A GIVEN TEMPERATURE AND DENSITY.
C
C           ALSO OUTPUTS INVERTED METASTABLE RATE MATRIX.
C
C  CALLING PROGRAM:  ADAS205/ADAS206
C
C  SUBROUTINE:
C
C  INPUT :  (I*4)  NDMET   = MAXIMUM NUMBER OF METASTABLE LEVELS ALLOWED
C
C  INPUT :  (I*4)  NMET    = NUMBER OF METASTABLE LEVELS
C
C  INPUT :  (R*8)  CRED(,) = MATRIX  OF  TRANSITION   RATES   BETWEEN
C                            METASTABLE LEVELS.
C                            (UNITS: SEC-1)
C                            VALUES FOR GIVEN TEMPERATURE AND DENSITY.
C                            1st DIMENSION: METASTABLE LEVEL INDEX
C                            2nd DIMENSION: METASTABLE LEVEL INDEX
C
C  OUTPUT:  (R*8)  RHS()   = GENERAL MATRIX SOLUTION WORK SPACE:
C                            USED IN SOLUTION OF 'NMET-1' LINEAR EQNS.
C                                          A.X=B
C                            INPUT TO XXMINV: RIGHT HAND SIDE VECTOR 'B'
C                             (RHS(IM) = -(RATE FROM LEVEL 'IM+1' TO 1))
C                             (UNITS: SEC-1)
C                            OUTPUT FROM XXMINV: SOLUTION VECTOR 'X'
C                             (RHS(IM) = POPULATION OF LEVEL 'IM+1')
C                            VALUES FOR GIVEN TEMPERATURE AND DENSITY.
C                            DIMENSION: METASTABLE LEVEL - 1
C  OUTPUT:  (R*8)  CRMAT(,)= INVERTED  METASTABLE  LEVEL   RATE   MATRIX
C                            COVERING ALL TRANSITIONS BETWEEN METASTABLE
C                            LEVELS EXCEPT THOSE INVOLVING LEVEL 1.
C                            VALUES FOR GIVEN TEMPERATURE AND DENSITY.
C                            BEFORE INPUT  TO   XXMINV: NOT INVERTED
C                            AFTER  OUTPUT FROM XXMINV: AS-ABOVE
C                            1st DIMENSION: METASTABLE LEVEL INDEX - 1
C                            2nd DIMENSION: METASTABLE LEVEL INDEX - 1
C
C  OUTPUT:  (R*8)  STKM()  = METASTABLE LEVEL POPULATION MATRIX.
C                            VALUES FOR GIVEN TEMPERATURE AND DENSITY.
C                            DIMENSION: METASTABLE LEVEL INDEX
C
C           (L*4)  LSOLVE  = PARAMETER = .TRUE.
C                                    => USE 'XXMINV' TO SOLVE A SET  OF
C                                       LINEAR EQUATIONS A.X = B, WHERE
C                                       A,X,B ARE MATRICES/VECTORS AND:
C                                        A='CRMAT(,)' INPUT  TO   XXMINV
C                                        B='RHS()'    INPUT  TO   XXMINV
```

```
C                                       X='RHS()'    OUTPUT FROM XXMINV
C
C           (I*4)  NMET1   = 'NMET - 1'
C           (I*4)  IM      = METASTABLE LEVEL ARRAY INDEX
C           (I*4)  IM1     = METASTABLE LEVEL ARRAY INDEX
C           (I*4)  IM2     = METASTABLE LEVEL ARRAY INDEX
C
C           (R*8)  DMINT   = +1 or -1 DEPENDING ON WHETHER THE NUMBER OF
C                            ROW   INTERCHANGES   WAS   EVEN   OR   ODD,
C                            RESPECTIVELY, WHEN INVERTING A MATRIX USING
C                            'XXMINV'.
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          XXMINV     ADAS      INVERTS MATRIX AND SOLVES EQUATIONS.
C
C NOTE:
C        THE SOLUTION OF METASTABLE POPULATIONS GIVEN BELOW IS BASED ON
C        METASTABLE LEVEL 1 HAVING A POPULATION OF UNITY (1.0).
C
C        IF:     m  =  number of metastable levels - 1
C
C            R(mxm) = Rate matrix (sec-1) covering transistions between
C                     all possible pairs of metastable levels (except 1)
C                     row   : final   level
C                     column: initial level
C                     (R(mxm)   = 'CRMAT(,)' on input  to   XXMINV)
C                     (R-1(mxm) = 'CRMAT(,)' on output from XXMINV)
C            V(m)   = Rate vector (sec-1) covering transistions between
C                     each metastable level (except 1) and met. level 1
C                     ( = 'RHS()' on input  to   XXMINV)
C            P(m)   = Metastable level populations - levels 2 -> 'NMET'
C                     ( = 'RHS()' on output from XXMINV)
C
C           Therefore:  R(mxm).P(m) = V(m)
C
C            =>         P(m) = R-1(mxm).V(m)
C
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 4569
C
C DATE:    09/10/90
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      LOGICAL    LSOLVE
C-----------------------------------------------------------------------
      INTEGER             NDMET,       NMET
      REAL*8              CRED(NDMET,NDMET),        CRMAT(NDMET,NDMET)
      REAL*8              RHS(NDMET),  STKM(NDMET)
```

3.110 bxpopm: Subroutine bxpopm from library adas2xx

```
      SUBROUTINE BXPOPM( NDTEM  , NDDEN , NDMET , NDLEV ,
     &                   MAXT   , MAXD  , NMET  ,
     &                            DENSA , IMETR ,
     &                            LRSEL , LHSEL ,
     &                            RATIA , RATHA ,
     &                   STCKM  , STVRM , STVHM ,
     &                   POPAR
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: BXPOPM *********************
C
C  PURPOSE: TO CONSTRUCT METASTABLE LEVEL POPULATIONS.
C
C  CALLING PROGRAM:  ADAS205/ADAS206
C
C  SUBROUTINE:
C
C  INPUT :  (I*4)  NDTEM   = MAXIMUM NUMBER OF TEMPERATURES ALLOWED
C  INPUT :  (I*4)  NDDEN   = MAXIMUM NUMBER OF DENSITIES ALLOWED
C  INPUT :  (I*4)  NDMET   = MAXIMUM NUMBER OF METASTABLE LEVELS ALLOWED
C  INPUT :  (I*4)  NDLEV   = MAXIMUM NUMBER OF ENERGY LEVELS ALLOWED
C
C  INPUT :  (I*4)  MAXT    = NUMBER OF INPUT TEMPERATURES ( 1 ->'NDTEM')
C  INPUT :  (I*4)  MAXD    = NUMBER OF INPUT DENSITIES ( 1 ->'NDDEN')
C  INPUT :  (I*4)  NMET    = NUMBER OF METASTABLES LEVELS ( 1 ->'NDMET')
C
C  INPUT :  (R*8)  DENSA() = ELECTRON DENSITIES  (UNITS: CM-3)
C  INPUT :  (I*4)  IMETR() = INDEX OF METASTABLE IN COMPLETE LEVEL LIST
C                            (ARRAY SIZE = 'NDMET' )
C
C  INPUT :  (L*4)  LRSEL   = .TRUE.  => FREE ELECTRON RECOMBINATION
C                                       REQUESTED.
C                          = .FALSE. => FREE ELECTRON RECOMBINATION
C                                       NOT REQUESTED.
C  INPUT :  (L*4)  LHSEL   = .TRUE.  => CHARGE TRANSFER FROM NEUTRAL
C                                       HYDROGREN REQUESTED.
C                          = .FALSE. => CHARGE TRANSFER FROM NEUTRAL
C                                       HYDROGREN NOT REQUESTED.
C
C  INPUT :  (R*8)  RATIA() = RATIO ( N(Z+1)/N(Z)  STAGE ABUNDANCIES )
C  INPUT :  (R*8)  RATHA() = RATIO (NEUTRAL H DENSITY/ELECTRON DENSITY)
C
C  INPUT :  (R*8)  STCKM(,,)= METASTABLE POPULATIONS STACK:
C                              1st DIMENSION: METASTABLE INDEX
C                              2nd DIMENSION: TEMPERATURE INDEX
C                              3rd DIMENSION: DENSITY INDEX
C  INPUT :  (R*8)  STVRM(,,)= METASTABLE LEVEL:
C                             FREE-ELECTRON RECOMBINATION COEFFICIENTS
C                             (UNITS* CM**3/SEC-1)
C                              1st DIMENSION: METASTABLE INDEX
C                              2nd DIMENSION: TEMPERATURE INDEX
C                              3rd DIMENSION: DENSITY INDEX
C  INPUT :  (R*8)  STVHM(,,)= METASTABLE LEVEL:
C                             CHARGE-EXCHANGE RECOMBINATION COEFFICIENTS
C                             (UNITS* CM**3/SEC-1)
C                              1st DIMENSION: METASTABLE INDEX
C                              2nd DIMENSION: TEMPERATURE INDEX
C                              3rd DIMENSION: DENSITY INDEX
```

```
C
C  OUTPUT:  (R*8)  POPAR(,,)= LEVEL POPULATIONS
C                              1st DIMENSION: LEVEL INDEX
C                              2nd DIMENSION: TEMPERATURE INDEX
C                              3rd DIMENSION: DENSITY INDEX
C                              (ON OUTPUT CONTAINS POPULATIONS FOR
C                               METASTABLE LEVELS ONLY.)
C
C           (R*8) DCOEF     = DENSITY MULTIPLIED BY RELEVANT RATIOS  FOR
C                             CALCULATING RECOMBINATION CONTRIBUTIONS.
C
C           (I*4) IT        = TEMPERATURE ARRAY INDEX
C           (I*4) IN        = DENSITY ARRAY INDEX
C           (I*4) IM        = METASTABLE LEVEL ARRAY INDEX
C
C
C ROUTINES: NONE
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 4569
C
C DATE:    09/10/90
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IMETR(NDMET),             MAXD,        MAXT
      INTEGER             NDDEN,       NDLEV,       NDMET,       NDTEM
      INTEGER             NMET
      LOGICAL             LHSEL,       LRSEL
      REAL*8              DENSA(NDDEN)
      REAL*8              POPAR(NDLEV,NDTEM,NDDEN), RATHA(NDDEN)
      REAL*8              RATIA(NDDEN)
      REAL*8              STCKM(NDMET,NDTEM,NDDEN)
      REAL*8              STVHM(NDMET,NDTEM,NDDEN)
      REAL*8              STVRM(NDMET,NDTEM,NDDEN)
```

3.111 bxpopo: Subroutine bxpopo from library adas2xx

```
      SUBROUTINE BXPOPO( NDTEM  , NDDEN , NDMET , NDLEV ,
     &                   MAXT   , MAXD  , NMET  , NORD  ,
     &                            DENSA , IMETR , IORDR ,
     &                            LRSEL , LHSEL ,
     &                            RATIA , RATHA ,
     &                   STACK  , STVR  , STVH  ,
     &                   POPAR
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: BXPOPO *********************
C
C  PURPOSE: TO CONSTRUCT ORDINARY/NON-METASTABLE LEVEL POPULATIONS.
C
C  CALLING PROGRAM:  ADAS205/ADAS206
C
C  SUBROUTINE:
C
C  INPUT :  (I*4)  NDTEM   = MAXIMUM NUMBER OF TEMPERATURES ALLOWED
C  INPUT :  (I*4)  NDDEN   = MAXIMUM NUMBER OF DENSITIES ALLOWED
C  INPUT :  (I*4)  NDMET   = MAXIMUM NUMBER OF METASTABLE LEVELS ALLOWED
C  INPUT :  (I*4)  NDLEV   = MAXIMUM NUMBER OF ENERGY LEVELS ALLOWED
C
C  INPUT :  (I*4)  MAXT    = NUMBER OF INPUT TEMPERATURES ( 1 ->'NDTEM')
C  INPUT :  (I*4)  MAXD    = NUMBER OF INPUT DENSITIES ( 1 ->'NDDEN')
C  INPUT :  (I*4)  NMET    = NUMBER OF METASTABLES LEVELS ( 1 ->'NDMET')
C  INPUT :  (I*4)  NORD    = NUMBER OF ORDINARY LEVELS ( 1 ->'NDLEV')
C
C  INPUT :  (R*8)  DENSA() = ELECTRON DENSITIES  (UNITS: CM-3)
C  INPUT :  (I*4)  IMETR() = INDEX OF METASTABLE IN COMPLETE LEVEL LIST
C                            (ARRAY SIZE = 'NDMET' )
C  INPUT :  (I*4)  IORDR() =INDEX OF ORDINARY EXCITED LEVELS IN COMPLETE
C                           LEVEL LIST.
C                           (ARRAY SIZE = 'NDLEV' )
C
C  INPUT :  (L*4)  LRSEL   = .TRUE.  => FREE ELECTRON RECOMBINATION
C                                       REQUESTED.
C                          = .FALSE. => FREE ELECTRON RECOMBINATION
C                                       NOT REQUESTED.
C  INPUT :  (L*4)  LHSEL   = .TRUE.  => CHARGE TRANSFER FROM NEUTRAL
C                                       HYDROGREN REQUESTED.
C                          = .FALSE. => CHARGE TRANSFER FROM NEUTRAL
C                                       HYDROGREN NOT REQUESTED.
C
C  INPUT :  (R*8)  RATIA() = RATIO ( N(Z+1)/N(Z)  STAGE ABUNDANCIES )
C  INPUT :  (R*8)  RATHA() = RATIO (NEUTRAL H DENSITY/ELECTRON DENSITY)
C
C  INPUT : (R*4) STACK(,,,)= ORDINARY EXCITED LEVEL POPULAT'N DEPENDENCE
C                           ON METASTABLE LEVEL.
C                            1st DIMENSION: ORDINARY LEVEL INDEX
C                            2nd DIMENSION: METASTABLE INDEX
C                            3rd DIMENSION: TEMPERATURE INDEX
C                            4th DIMENSION: DENSITY INDEX
C  INPUT : (R*8) STVR(,,)  = ORDINARY EXCITED LEVEL:
C                            FREE-ELECTRON RECOMBINATION COEFFICIENTS
C                            (UNITS* CM**3/SEC-1)
C                            1st DIMENSION: ORDINARY LEVEL INDEX
C                            2nd DIMENSION: TEMPERATURE INDEX
C                            3rd DIMENSION: DENSITY INDEX
```

```
C  INPUT : (R*8) STVH(,,)  = ORDINARY EXCITED LEVEL:
C                             CHARGE-EXCHANGE RECOMBINATION COEFFICIENTS
C                             (UNITS* CM**3/SEC-1)
C                             1st DIMENSION: ORDINARY LEVEL INDEX
C                             2nd DIMENSION: TEMPERATURE INDEX
C                             3rd DIMENSION: DENSITY INDEX
C
C  I/O   :  (R*8)  POPAR(,,)= LEVEL POPULATIONS
C                              1st DIMENSION: LEVEL INDEX
C                              2nd DIMENSION: TEMPERATURE INDEX
C                              3rd DIMENSION: DENSITY INDEX
C                               ON INPUT : CONTAINS POPULATIONS FOR
C                                          METASTABLE LEVELS ONLY.
C                               ON OUTPUT: CONTAINS POPULATIONS FOR
C                                          ALL LEVELS.
C
C           (R*8) DCOEF     = DENSITY MULTIPLIED BY RELEVANT RATIOS  FOR
C                             CALCULATING RECOMBINATION CONTRIBUTIONS.
C
C           (I*4) IT        = TEMPERATURE ARRAY INDEX
C           (I*4) IN        = DENSITY ARRAY INDEX
C           (I*4) IO        = ORDINARY LEVEL ARRAY INDEX
C           (I*4) IM        = METASTABLE LEVEL ARRAY INDEX
C
C
C ROUTINES: NONE
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 4569
C
C DATE:    09/10/90
C
C UPDATE:  20/05/93- P BRIDEN: STACK ARRAY CHANGED FROM REAL*8 -> REAL*4
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IMETR(NDMET),             IORDR(NDLEV)
      INTEGER             MAXD,        MAXT,        NDDEN,       NDLEV
      INTEGER             NDMET,       NDTEM,       NMET,        NORD
      LOGICAL             LHSEL,       LRSEL
      REAL*8              DENSA(NDDEN)
      REAL*8              POPAR(NDLEV,NDTEM,NDDEN), RATHA(NDDEN)
      REAL*8              RATIA(NDDEN)
      REAL                STACK(NDLEV,NDMET,NDTEM,NDDEN)
      REAL*8              STVH(NDLEV,NDTEM,NDDEN)
      REAL*8              STVR(NDLEV,NDTEM,NDDEN)
```

3.112 bxrate: Subroutine bxrate from library adas2xx

```
      SUBROUTINE BXRATE( NDTEM  , NDTRN  , GSCALE ,
     &                   NTIN   , TIN    , GAMIN  ,
     &                   NTOUT  , TOUT   ,
     &                   ICNT   , ITRN   ,
     &                   RATE   , DRATE  ,
     &                   LTRNG
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: BXRATE *********************
C
C  PURPOSE: TO CALCULATE THE EXCITATION AND DE-EXCITATION RATE COEFFICI-
C           ENTS FOR A SET OF INPUT TEMPERATURES 'TOUT' & TRANSITIONS OF
C           A SPECIFIED TYPE (ELECTRON OR PROTON IMPACT).
C
C           TRANSITION TYPE SELECTED VIA 'ICNT & ITRN'.
C
C           INPUT RATE COEFFICIENTS 'RATE' & 'DRATE'  ASSUME  THAT  THE
C           GAMMA VALUE IS UNITY, AND ARE GIVEN FOR THE TEMPERATURES IN
C           'TOUT'. THE GAMMA VALUES 'GAMIN' ARE FOR  THE  TEMPERARTURE
C           ARRAY 'TIN'.  SPLINES ARE USED  TO  EXTRAPOLATE/INTERPOLATE
C           THE GAMMA VALUES INTO THE 'TOUT' ARRAY  AND  THESE USED  TO
C           CALCULATE THE CORRECT RATE COEFFICIENTS.
C
C           SPLINE IS CARRIED OUT USING LOG(GAMMA VALUES)
C
C  CALLING PROGRAM:  ADAS205/ADAS206
C
C  SUBROUTINE:
C
C  INPUT :  (I*4)  NDTEM   = MAXIMUM NUMBER OF TEMPERATURES ALLOWED
C  INPUT :  (I*4)  NDTRN   = MAXIMUM NUMBER OF TRANSITIONS ALLOWED
C  INPUT :  (I*4)  GSCALE  = SCALING FACTOR FOR OUTPUT GAMMA VALUES
C
C  INPUT :  (I*4)  NTIN    = NUMBER OF TEMPERATURES REPRESENTED IN THE
C                            INPUT DATA SET.
C  INPUT :  (R*8)  TIN()   = TEMPERATURES REPRESENTED IN INPUT DATA SET
C  INPUT :  (R*8)  GAMIN(,)= GAMMA VALUES REPRESENTED IN INPUT DATA SET
C                            1st DIMENSION: TEMPERATURE INDEX ('TIN')
C                            2nd DIMENSION: TRANSITION INDEX
C                                           (SEE: 'ITRN()')
C
C  INPUT :  (I*4)  NTOUT   = NUMBER OF ISPF SELECTED TEMPERATURES FOR
C                            OUTPUT.
C  INPUT :  (R*8)  TOUT()  = ISPF SELECTED TEMPERATURES FOR OUTPUT.
C
C  INPUT :  (I*4)  ICNT    = NUMBER OF SELECTED TRANSITIONS
C  INPUT :  (I*4)  ITRN()  = INDEX VALUES IN MAIN TRANSITION ARRAY WHICH
C                            REPRESENT TRANSITIONS OF THE SELECTED TYPE.
C                            USED TO SELECT APPROPRIATE GAMMA VALUES FOR
C                            TRANSITION TYPE.
C
C  I/O   :  (R*8)  RATE(,) = EXCITATION RATE COEFFS (cm**3/s)
C                            INPUT : UNIT GAMMA VALUES
C                            OUTPUT: TRUE VALUES
C                            1st DIMENSION: TEMPERATURE INDEX ('TOUT')
C                            2nd DIMENSION: TRANSITION INDEX
C  I/O   :  (R*8)  DRATE(,)= DE-EXCIT'N RATE COEFFS (cm**3/s)
C                            INPUT : UNIT GAMMA VALUES
```

```
C                             OUTPUT: TRUE VALUES
C                             1st DIMENSION: TEMPERATURE INDEX ('TOUT')
C                             2nd DIMENSION: TRANSITION INDEX
C
C  OUTPUT:  (L*4)  LTRNG() = .TRUE. => TEMPERATURE VALUES WITHIN RANGE
C                                      READ FROM INPUT COPASE DATA SET.
C                             = .FALSE.=>TEMPERATURE VALUE NOT WITHIN RANGE
C                                      READ FROM INPUT COPASE DATA SET.
C                             1st DIMENSION: TEMPERATURE INDEX.
C
C
C           (I*4)  NTDSN   = PARAMETER = MAXIMUM NUMBER OF TEMPERATURES
C                                        ALLOWED IN INPUT DATA SET = 8
C           (I*4)  NLTEM   = PARAMETER = MUST BE >= 'NDTEM'
C
C           (I*4)  GZERO   = PARAMETER = IF 'GAMIN(1,) < GZERO' THEN ALL
C                                        THE 'RATE' AND  'DRATE'  VALUES
C                                        FOR THE  GIVEN  TRANSITION  ARE
C                                        SAID TO BE ZERO.
C
C           (I*4)  IOPT    = SPLINE END CONDITIONS/EXTRAPOLATION CONTROL
C                            SWITCH - SEE 'XXSPLE'
C                            I.E. DEFINES THE BOUNDARY DERIVATIVES.
C                            (VALID VALUES = 0, 1, 2, 3, 4)
C           (I*4)  ITRAN   = APPROPRIATE TRANSITION INDEX FOR 'GAMIN(,)'
C           (I*4)  IC      = TRANSITION ARRAY INDEX
C           (I*4)  IT      = TEMPERATURE ARRAY INDEX
C
C           (R*8)  GAMMA   = SPLINED GAMMA VALUE FOR GIVEN TEMPERATURE
C                            (FROM 'TOUT()') AND TRANSITION.
C           (R*8)  DYIN()  = INTERPOLATED DERIVATIVES
C                            DIMENSION: TEMPERATURE INDEX ('TIN()')
C
C           (L*4)  LSETX   = .TRUE.  => X-AXES ('TIN()' VALUES) NEED TO
C                                       SET IN 'XXSPLE'.
C                            .FALSE. => X-AXES ('TIN()' VALUES) HAVE
C                                       BEEN SET IN 'XXSPLE'.
C                            (NOTE: 'LSETX' IS RESET BY 'XXSPLE')
C
C           (R*8)  LGIN()  = LOG ( 'GAMIN(,)' ) FOR GIVEN TRANSITION
C                            DIMENSION: TEMPERATURE INDEX ('TIN()')
C           (R*8)  LGOUT() = LOG ( SPLINED GAMMA VALUES )
C                            DIMENSION: TEMPERATURE INDEX ('TOUT()' )
C
C
C ROUTINES:
C           ROUTINE    SOURCE    BRIEF DESCRIPTION
C           ------------------------------------------------------------
C           XXSPLE     ADAS      SPLINE SUBROUTINE (WITH EXTRAP. INFO)
C
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 4569
C
C DATE:    09/10/90
C
C UPDATE:  31/01/91 - PE BRIDEN - ADAS91 - INTRODUCED 'LTRNG'
C                                        - REPLACED XXSPLN WITH XXSPLE
C
C UPDATE:  26/03/91 - PE BRIDEN - ADAS91 - IF 'GAMIN(1,)' <='GZERO' THEN
```

```
C                                          SET 'RATE' AND 'DRATE' TO 0.0
C                                          FOR ALL TEMPERATURE VALUES.
C                                           * INCLUDED  FOR  LATER  USE.
C                                             AT PRESENT 'BXDATA'  MAKES
C                                             SURE 'GAMIN' HAS A MINIMUM
C                                             VALUE OF 1.00D-30. *
C
C UPDATE:  11/12/91 - PE BRIDEN - ADAS91 -NLTEM INCREASED FROM 20 to 101
C
C UPDATE:  20/05/93 - PE BRIDEN - ADAS91 -NTDSN INCREASED FROM 8 to 14
C                                         (REFLECTS CHANGES TO BXDATA)
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             ICNT,         ITRN(NDTRN), NDTEM,       NDTRN
      INTEGER             NTIN,         NTOUT
      LOGICAL             LTRNG(NDTEM)
      REAL*8              DRATE(NDTEM,NDTRN),       GAMIN(NTDSN,NDTRN)
      REAL*8              GSCALE,       RATE(NDTEM,NDTRN)
      REAL*8              TIN(NTDSN),   TOUT(NDTEM)
```

3.113 bxrcom: Subroutine bxrcom from library adas2xx

```
      SUBROUTINE BXRCOM( NDTEM  , NDTRN  , NDLEV  ,
     &                   NTIN   , TIN    , RCIN   ,
     &                   NTOUT  , TOUT   ,
     &                   ICNT   , ITRN   , ICLEV  ,
     &                   RCOUT  , LTRNG
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: BXRCOM *********************
C
C  PURPOSE: TO ESTABLISH RECOMBINATION RATE COEFFICIENTS  FOR  A SET  OF
C           TEMPERATURES GIVEN BY THE ARRAY 'TOUT()' USING CUBIC SPLINES
C           ON  A SET OF RATE  COEFFICIENTS  COVERING  THE  TEMPERATURES
C           GIVEN BY THE ARRAY 'TIN()'.
C
C           RECOMBINATION TYPE IS SELECTED VIA 'ICNT' & 'ITRN'
C
C           RATE COEFFICIENTS ARE GIVEN FOR A NUMBER OF CAPTURING LEVELS
C           AND THE ARRAY 'RCOUT(,)' REPRESENTS  COEFFTS.  FOR COMBINAT-
C           IONS OF TEMPERATURE AND CAPTURING LEVEL INDEX.
C
C           SPLINE IS CARRIED OUT USING LOG(RATE COEFFICIENT VALUES)
C
C  CALLING PROGRAM:  ADAS205/ADAS206
C
C  SUBROUTINE:
C
C  INPUT :  (I*4)  NDTEM   = MAXIMUM NUMBER OF TEMPERATURES ALLOWED
C  INPUT :  (I*4)  NDTRN   = MAXIMUM NUMBER OF RECOMBINATIONS ALLOWED
C  INPUT :  (I*4)  NDLEV   = MAXIMUM NUMBER OF ENERGY LEVELS ALLOWED
C
C  INPUT :  (I*4)  NTIN    = NUMBER OF TEMPERATURES REPRESENTED IN THE
C                            INPUT DATA SET.
C  INPUT :  (R*8)  TIN()   = TEMPERATURES REPRESENTED IN INPUT DATA SET
C  INPUT :  (R*8)  RCIN(,) = RATE COEFF. REPRESENTED IN INPUT DATA SET
C                            1st DIMENSION: TEMPERATURE INDEX ('TIN')
C                            2nd DIMENSION: RECOMBINATION INDEX
C                                           (SEE: 'ITRN()')
C
C  INPUT :  (I*4)  NTOUT   = NUMBER OF ISPF SELECTED TEMPERATURES FOR
C                            OUTPUT.
C  INPUT :  (R*8)  TOUT()  = ISPF SELECTED TEMPERATURES FOR OUTPUT.
C
C  INPUT :  (I*4)  ICNT    = NUMBER OF SELECTED RECOMBINATIONS
C  INPUT :  (I*4)  ITRN()  = INDEX VALUES IN MAIN TRANSITION ARRAY WHICH
C                            REPRESENT RECOMBINASTION  OF THE  SELECTED
C                            TYPE
C                            USED TO SELECT APPROPRIATE RATE COEFFTS FOR
C                            RECOMBINATION TYPE.
C  INPUT :  (I*4)  ICLEV() = CAPTURING LEVELS INDICES.
C                            DIMENSION: 'TRANSITION'/RECOMBINATION INDEX
C
C  OUTPUT:  (R*8)  RCOUT(,)= SPLINED RECOMBINATION RATE COEFFT. VALUES.
C                            1st DIMENSION: TEMPERATURE INDEX ('TOUT')
C                            2nd DIMENSION: CAPTURING LEVEL INDEX.
C
C  OUTPUT:  (L*4)  LTRNG() = .TRUE. => TEMPERATURE VALUES WITHIN RANGE
C                                      READ FROM INPUT COPASE DATA SET.
C                          = .FALSE.=>TEMPERATURE VALUE NOT WITHIN RANGE
```

```
C                                   READ FROM INPUT COPASE DATA SET.
C                             1st DIMENSION: TEMPERATURE INDEX.
C
C
C           (I*4)  NTDSN   = PARAMETER = MAXIMUM NUMBER OF TEMPERATURES
C                                        ALLOWED IN INPUT DATA SET = 8
C           (I*4)  NLTEM   = PARAMETER = MUST BE >= 'NDTEM'
C
C           (I*4)  IOPT    = SPLINE END CONDITIONS/EXTRAPOLATION CONTROL
C                            SWITCH - SEE 'XXSPLE'
C                            I.E. DEFINES THE BOUNDARY DERIVATIVES.
C                            (VALID VALUES = 0, 1, 2, 3, 4)
C           (I*4)  IRECMB  = APPROPRIATE RECOMBINATN INDEX FOR 'RCIN(,)'
C           (I*4)  ICAP    = CAPTURING LEVEL INDEX BEING ASSESSED.
C           (I*4)  IC      = RECOMBINATION ARRAY INDEX
C           (I*4)  IT      = TEMPERATURE ARRAY INDEX
C
C           (R*8)  DYIN()  = INTERPOLATED DERIVATIVES
C                            DIMENSION: TEMPERATURE INDEX ('TIN()')
C
C           (L*4)  LSETX   = .TRUE.  => X-AXES ('TIN()' VALUES) NEED TO
C                                       SET IN 'XXSPLE'.
C                            .FALSE. => X-AXES ('TIN()' VALUES) HAVE
C                                       BEEN SET IN 'XXSPLE'.
C                            (NOTE: 'LSETX' IS RESET BY 'XXSPLE')
C
C           (R*8)  LRCIN() = LOG ( 'RCIN(,)' ) FOR GIVEN CAPTURING LEVEL
C                            DIMENSION: TEMPERATURE INDEX ('TIN()')
C           (R*8)  LRCOUT()= LOG ( SPLINED RECOMBINATION RATE COEFTS )
C                            DIMENSION: TEMPERATURE INDEX ('TOUT()' )
C
C
C ROUTINES:
C           ROUTINE    SOURCE    BRIEF DESCRIPTION
C           ------------------------------------------------------------
C           XXSPLE     ADAS      SPLINE SUBROUTINE (WITH EXTRAP. INFO)
C
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 4569
C
C DATE:    09/10/90
C
C UPDATE:  31/01/91 - PE BRIDEN - ADAS91 - INTRODUCED 'LTRNG'
C                                        - REPLACED XXSPLN WITH XXSPLE
C
C UPDATE:  11/12/91 - PE BRIDEN - ADAS91 -NLTEM INCREASED FROM 20 to 101
C
C UPDATE:  10/06/92 - PE BRIDEN - ADAS91 -CORRECT ERROR - CHANGED
C                                         'ICAP=ICLEV(IC)' TO
C                                         'ICAP=ICLEV(IRECMB)'
C
C UPDATE:  20/05/93 - PE BRIDEN - ADAS91 -NTDSN INCREASED FROM 8 to 14
C                                         (REFLECTS CHANGES TO BXDATA)
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             ICLEV(NDTRN),             ICNT
      INTEGER             ITRN(NDTRN), NDLEV,       NDTEM,       NDTRN
```

```
INTEGER             NTIN,         NTOUT
LOGICAL             LTRNG(NDTEM)
REAL*8              RCIN(NTDSN,NDTRN),         RCOUT(NDTEM,NDLEV)
REAL*8              TIN(NTDSN),   TOUT(NDTEM)
```

3.114 bxsetp: Subroutine bxsetp from library adas2xx

```
       SUBROUTINE BXSETP( IZ0    , IZ     ,
     &                    NDLEV  , IL     , ICNTE ,
     &                    CSTRGA , ISA    , ILA   , XJA  ,
     &                    STRGA
     &                  )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: BXSETP *********************
C
C  PURPOSE:  TO SET UP PARAMETERS IN THE SHARED POOLED FOR PANEL DISPLAY
C
C  CALLING PROGRAM: ADAS205/ADAS206
C
C  DATA:
C           DATA IS OBTAINED VIA SUBROUTINE 'BXDATA'
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  IZ0     =         NUCLEAR CHARGE READ
C  INPUT : (I*4)  IZ      =  RECOMBINED ION CHARGE READ
C
C  INPUT : (I*4)  NDLEV   = MAXIMUM NUMBER OF ENERGY LEVELS ALLOWED
C  INPUT : (I*4)  IL      = INPUT DATA FILE: NUMBER OF ENERGY LEVELS
C  INPUT : (I*4)  ICNTE   = NUMBER OF ELECTRON IMPACT TRANSITIONS
C
C  INPUT : (C*18) CSTRGA()= NOMENCLATURE/CONFIGURATION FOR LEVEL 'IA()'
C  INPUT : (I*4)  ISA()   = MULTIPLICITY FOR LEVEL 'IA()'
C                           NOTE: (ISA-1)/2 = QUANTUM NUMBER (S)
C  INPUT : (I*4)  ILA()   = QUANTUM NUMBER (L) FOR LEVEL 'IA()'
C  INPUT : (R*8)  XJA()   = QUANTUM NUMBER (J-VALUE) FOR LEVEL 'IA()'
C                           NOTE: (2*XJA)+1 = STATISTICAL WEIGHT
C
C  OUTPUT: (C*22) STRGA() = LEVEL DESIGNATIONS
C
C          (C*8)  F6      = PARAMETER = 'VREPLACE'
C
C          (I*4)  ILEN    = LENGTH, IN BYTES, OF ISPF DIALOG VARIABLES
C          (I*4)  ILEV    = ARRAY COUNTER FOR LEVEL INDEX
C          (I*4)  J       = VALUE OF QUANTUM NUMBER L + 1
C          (I*4)  LFPOOL  = NO. OF LEVEL STRINGS SENT TO FUNCTION POOL
C
C          (C*2)  SZ0     =         NUCLEAR CHARGE READ
C          (C*2)  SZ      =  RECOMBINED ION CHARGE READ
C          (C*4)  SCNTE   =  NUMBER OF ELECTRON IMPACT TRANSITIONS
C          (C*4)  SIL     =  NUMBER OF ENERGY LEVELS
C          (C*1)  CONFIG()= QUANTUM NUMBER (L) LETTERS
C                           DIMENSION: QUANTUM NUMBER L + 1
C          (C*8)  CHA()   = FUNCTION POOL NAMES: CHARGE VALUES
C          (C*8)  CHB()   = FUNCTION POOL NAMES: LEVEL DESIGNATIONS <=99
C          (I*4)  PIPEOU  = PARAMETER = UNIT NUMBER FOR OUTPUT TO PIPE
C
C
C ROUTINES:
C           ROUTINE     SOURCE      BRIEF DESCRIPTION
C           ------------------------------------------------------------
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
```

```
C          JET EXT. 4569
C
C DATE:    09/10/90
C
C UPDATE:  26/06/92 - ADAS91 PEB: PLACED A LIMIT OF 99 ON THE NUMBER OF
C                                 LEVEL STRINGS SENT TO FUNCTION POOL.
C          15/04/93 - AB: CONVERTED TO WRITE TO IDL VIA PIPE INSTEAD
C                         OF ISPF FUNCTION POOL.
C          14/03/95 - L. JALOTA : INCREASED CSTRG TO 18 BYTES IN LINE WITH
C   NEW ADF04 FILE FORMATS AND READ ROUTINES.
C          05/04/95 - TIM HAMMOND : ADDED SEVERAL CALLS TO XXFLSH ROUTINE
C                                   TO CLEAR THE PIPE AFTER IT HAS BEEN
C                                   WRITTEN TO.
C V 1.5:   23/11/95 - TIM HAMMOND : ALTERED FORMAT 1001 TO I4 FROM I3
C                                   AND INCREASED SCNTE AND SIL FROM A3
C                                   TO A4 TO ALLOW FOR LARGER NUMBERS.
C
C VERSION: 1.6                              DATE: 29-04-2003
C MODIFIED: Martin O'Mullane
C - Remove the restriction of only sending LFPOOL (99)
C                 values of level string identifer to IDL. Send the
C                 complete level list and improve the data handling
C                 at the IDL level.
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      CHARACTER*18        CSTRGA(IL)
      CHARACTER*22        STRGA(NDLEV)
      INTEGER             ICNTE,       IL,          ILA(IL),     ISA(IL)
      INTEGER             IZ,          IZ0,         NDLEV
      REAL*8              XJA(IL)
```

3.115 bxstka: Subroutine bxstka from library adas2xx

```
      SUBROUTINE BXSTKA( NDLEV  , NDMET  ,
     &                   NORD   , NMET   ,
     &                   IORDR  , IMETR  ,
     &                   CMAT   , CC     ,
     &                   STCK
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: BXSTKA *********************
C
C  PURPOSE: TO STACK UP IN  'STCK'  THE NON-METASTABLE/ORDINARY EXCITED
C           LEVEL POPULATION DEPENDENCE ON METASTABLE LEVEL FOR A GIVEN
C           TEMPERATURE AND DENSITY.
C
C  CALLING PROGRAM:  ADAS205/ADAS206
C
C  SUBROUTINE:
C
C  INPUT :  (I*4)  NDLEV   = MAXIMUM NUMBER OF ENERGY LEVELS ALLOWED
C  INPUT :  (I*4)  NDMET   = MAXIMUM NUMBER OF METASTABLE LEVELS ALLOWED
C
C  INPUT :  (I*4)  NORD    = NUMBER OF ORDINARY EXCITED LEVELS
C  INPUT :  (I*4)  NMET    = NUMBER OF METASTABLE LEVELS
C
C  INPUT :  (I*4)  IMETR() = INDEX OF METASTABLE IN COMPLETE LEVEL LIST
C                            (ARRAY SIZE = 'NDMET' )
C  INPUT :  (I*4)  IORDR() =INDEX OF ORDINARY EXCITED LEVELS IN COMPLETE
C                           LEVEL LIST.
C                           (ARRAY SIZE = 'NDLEV' )
C
C  INPUT :  (R*8)  CMAT(,) = INVERTED   RATE   MATRIX   COVERING   ALL
C                           NON-METASTABLE/ORDINARY EXCITED LEVELS.
C                           (UNITS: SEC)
C                           VALUES FOR GIVEN TEMPERATURE AND DENSITY.
C                           1st DIMENSION: ORDINARY EXCITED LEVEL INDEX
C                           2nd DIMENSION: ORDINARY EXCITED LEVEL INDEX
C
C  INPUT :  (R*8)  CC(,)   = RATE MATRIX COVERING ALL TRANSITIONS
C                           (UNITS: SEC-1)
C                           VALUES FOR GIVEN TEMPERATURE AND DENSITY.
C                           1st DIMENSION: ENERGY LEVEL INDEX
C                           2nd DIMENSION: ENERGY LEVEL INDEX
C
C  OUTPUT:  (R*4)  STCK(,) = POPULATION MATRIX COVERING ALL NON-METAST-
C                           ABLE/ORDINARY EXCITED LEVELS AS FUNCTION
C                           OF METASTABLE INDEX.
C                           VALUES FOR GIVEN TEMPERATURE AND DENSITY.
C                           1st DIMENSION: ORDINARY EXCITED LEVEL INDEX
C                           2nd DIMENSION: METASTABLE LEVEL INDEX
C
C           (I*4)  IS1     = ORDINARY EXCITED LEVEL INDEX
C           (I*4)  IS2     = ORDINARY EXCITED LEVEL INDEX
C           (I*4)  IM      = METASTABLE LEVEL ARRAY INDEX
C
C           (R*8)  POP     = VARIABLE USED TO SUM POPULATION VALUES
C
C
C ROUTINES: NONE
C
```

```
C NOTE:
C        IF:     n  =  number of ordinary/non-metastable levels
C                m  =  number of metastable levels
C           Ro(nxn) = Rate matrix (sec-1) covering transistions between
C                     all possible pairs of ordinary levels.
C                     row   : final   level
C                     column: initial level
C                     (Inverse Ro-1(nxn) = 'CMAT(,)' )
C           Rm(nxm) = Rate matrix (sec-1) covering transistions between
C                     all combinations of ordinary and metastable level
C                     ( = 'CC(,)' - ordinary level part )
C           P(nxm)  = Population matrix giving the population dependence
C                     of each ordinary level on metastable level.
C                     ( = 'STCK(,)' )
C
C           Therefore:  Ro(nxn).P(nxm) = Rm(nxm)
C
C            =>          P(nxm)  = Ro-1(nxn).Rm(nxm)
C
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 4569
C
C DATE:    09/10/90
C
C UPDATE:  20/05/93 - P BRIDEN: STCK ARRAY CHANGED FROM REAL*8 -> REAL*4
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IMETR(NDMET),             IORDR(NDLEV)
      INTEGER             NDLEV,       NDMET,       NMET,        NORD
      REAL*8              CC(NDLEV,NDLEV),          CMAT(NDLEV,NDLEV)
      REAL                STCK(NDLEV,NDMET)
```

3.116 bxstkb: Subroutine bxstkb from library adas2xx

```
      SUBROUTINE BXSTKB( NDTEM  , NDLEV  ,
     &                   IT     , NORD   ,
     &                            IORDR  ,
     &                   CMAT   , VEC    ,
     &                   STV
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: BXSTKA *********************
C
C  PURPOSE: TO STACK UP IN  'STV'  THE  RECOMBINATION CONTRIBUTION  FOR
C           EACH  NON-METASTABLE/ORDINARY  EXCITED  LEVEL  FOR  A GIVEN
C           TEMPERATURE AND DENSITY.
C
C  CALLING PROGRAM:  ADAS205/ADAS206
C
C  SUBROUTINE:
C
C  INPUT :  (I*4)  NDTEM   = MAXIMUM NUMBER OF TEMPERATURES ALLOWED
C  INPUT :  (I*4)  NDLEV   = MAXIMUM NUMBER OF ENERGY LEVELS ALLOWED
C
C  INPUT :  (I*4)  IT      = INDEX DENOTING THE TEMPERATURE
C  INPUT :  (I*4)  NORD    = NUMBER OF ORDINARY EXCITED LEVELS
C
C  INPUT :  (I*4)  IORDR() =INDEX OF ORDINARY EXCITED LEVELS IN COMPLETE
C                           LEVEL LIST.
C                           (ARRAY SIZE = 'NDLEV' )
C
C  INPUT :  (R*8)  CMAT(,) = INVERTED   RATE   MATRIX   COVERING   ALL
C                           NON-METASTABLE/ORDINARY EXCITED LEVELS
C                           TRANSITIONS.
C                           (UNITS: SEC)
C                           VALUES FOR GIVEN TEMPERATURE AND DENSITY.
C                           1st DIMENSION: ORDINARY EXCITED LEVEL INDEX
C                           2nd DIMENSION: ORDINARY EXCITED LEVEL INDEX
C
C  INPUT :  (R*8)  VEC(,)  = RECOMBINATION RATE COEFFT. VALUES.
C                           (UNITS: CM**3/SEC-1)
C                           VALUES FOR GIVEN TEMPERATURE AND DENSITY.
C                           1st DIMENSION: TEMPERATURE INDEX ('IT')
C                           2nd DIMENSION: CAPTURING LEVEL INDEX
C
C  OUTPUT:  (R*8)  STV()   = RECOMBINATION  CONTRIBUTION  FOR  EACH
C                           NON-METASTABLE/ORDINARY EXCITED LEVELS.
C                           (UNITS: CM**3)
C                           VALUES FOR GIVEN TEMPERATURE AND DENSITY.
C                           DIMENSION: ORDINARY EXCITED LEVEL INDEX
C
C           (I*4)  IS1     = ORDINARY EXCITED LEVEL INDEX
C           (I*4)  IS2     = ORDINARY EXCITED LEVEL INDEX
C
C           (R*8)  COEF    = VARIABLE USED TO SUM COEFFICIENT VALUES
C
C
C ROUTINES: NONE
C
C NOTE:
C        IF:     n  =  number of ordinary/non-metastable levels
C            R(nxn) = Rate matrix (SEC-1) covering transistions between
```

```
C                     all possible pairs of ordinary levels.
C                     row   : final   level
C                     column: initial level
C                     (Inverse R-1(nxn) = 'CMAT(,)' )
C            V(n)   = Recombination rate vector (CM**3 SEC-1) covering
C                     all ordinary levels.
C                     ( = 'VEC()' - ordinary level part ).
C            S(n)   = Recombination contribution vector (CM**3) covering
C                     all ordinary levels ( = 'STV()' ).
C
C           Therefore:  R(nxn).S(n) = V(n)
C
C            =>         S(n)  = R-1(nxn).V(n)
C
C
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 4569
C
C DATE:    09/10/90
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IORDR(NDLEV),             IT,          NDLEV
      INTEGER             NDTEM,       NORD
      REAL*8              CMAT(NDLEV,NDLEV),        STV(NDLEV)
      REAL*8              VEC(NDTEM,NDLEV)
```

3.117 bxstkc: Subroutine bxstkc from library adas2xx

```
      SUBROUTINE BXSTKC( NDLEV  , NDMET  ,
     &                   NORD   , NMET   ,
     &                   IORDR  , IMETR  ,
     &                   CC     , STCK   ,
     &                   CRED
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: BXSTKC *********************
C
C  PURPOSE: TO STACK UP IN  'CRED'  THE TRANSITION RATE BETWEEN METASTA-
C           BLE LEVELS FOR A GIVEN TEMPERATURE STABLE LEVEL FOR A GIVEN
C           TEMPERATURE AND DENSITY.
C
C  CALLING PROGRAM:  ADAS205/ADAS206
C
C  SUBROUTINE:
C
C  INPUT :  (I*4)  NDLEV   = MAXIMUM NUMBER OF ENERGY LEVELS ALLOWED
C  INPUT :  (I*4)  NDMET   = MAXIMUM NUMBER OF METASTABLE LEVELS ALLOWED
C
C  INPUT :  (I*4)  NORD    = NUMBER OF ORDINARY EXCITED LEVELS
C  INPUT :  (I*4)  NMET    = NUMBER OF METASTABLE LEVELS
C
C  INPUT :  (I*4)  IMETR() = INDEX OF METASTABLE IN COMPLETE LEVEL LIST
C                            (ARRAY SIZE = 'NDMET' )
C  INPUT :  (I*4)  IORDR() =INDEX OF ORDINARY EXCITED LEVELS IN COMPLETE
C                           LEVEL LIST.
C                           (ARRAY SIZE = 'NDLEV' )
C
C  INPUT :  (R*8)  CC(,)   = RATE MATRIX COVERING ALL TRANSITIONS
C                           (UNITS: SEC-1)
C                           VALUES FOR GIVEN TEMPERATURE AND DENSITY.
C                           1st DIMENSION: ENERGY LEVEL INDEX
C                           2nd DIMENSION: ENERGY LEVEL INDEX
C  INPUT :  (R*4)  STCK(,) = POPULATION MATRIX COVERING ALL NON-METAST-
C                           ABLE/ORDINARY EXCITED LEVELS AS FUNCTION
C                           OF METASTABLE INDEX.
C                           VALUES FOR GIVEN TEMPERATURE AND DENSITY.
C                           1st DIMENSION: ORDINARY EXCITED LEVEL INDEX
C                           2nd DIMENSION: METASTABLE LEVEL INDEX
C
C  OUTPUT:  (R*8)  CRED(,) = MATRIX  OF  TRANSITION   RATES   BETWEEN
C                           METASTABLE LEVELS.
C                           (UNITS: SEC-1)
C                           VALUES FOR GIVEN TEMPERATURE AND DENSITY.
C                           1st DIMENSION: METASTABLE LEVEL INDEX
C                           2nd DIMENSION: METASTABLE LEVEL INDEX
C
C           (I*4)  IM1     = METASTABLE LEVEL ARRAY INDEX
C           (I*4)  IM2     = METASTABLE LEVEL ARRAY INDEX
C           (I*4)  IS      = ORDINARY EXCITED LEVEL INDEX
C
C
C ROUTINES: NONE
C
C NOTE:
C           CRED(IM1,IM2) = ( the transition rate from IM2 to IM1 )
C                                                +
```

```
C                           SUM( (the transistion rate from ordinary
C                                 level IS to IM1)  x (the population
C                                 in metastable level IM2 that excite
C                                 to oridinary level IS) )
C
C                           ABOVE SUM IS OVER ALL ORDINARY LEVELS.
C
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 4569
C
C DATE:    09/10/90
C
C UPDATE:  20/05/93 - P BRIDEN: STCK ARRAY CHANGED FROM REAL*8 -> REAL*4
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IMETR(NDMET),             IORDR(NDLEV)
      INTEGER             NDLEV,       NDMET,       NMET,        NORD
      REAL*8              CC(NDLEV,NDLEV),          CRED(NDMET,NDMET)
      REAL                STCK(NDLEV,NDMET)
```

3.118 bxstkd: Subroutine bxstkd from library adas2xx

```
      SUBROUTINE BXSTKD( NDTEM  , NDLEV  , NDMET  ,
     &                   IT     , NORD   , NMET   ,
     &                            IORDR  , IMETR  ,
     &                   CC     , STV    , VEC    ,
     &                   VRED
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: BXSTKD *********************
C
C  PURPOSE: TO STACK UP IN  'VRED'  THE RECOMBINATION RATE CONTRIBUTIONS
C           FOR  EACH  METASTABLE  LEVEL  FOR  A  GIVEN TEMPERATURE  AND
C           DENSITY.
C
C  CALLING PROGRAM:  ADAS205/ADAS206
C
C  SUBROUTINE:
C
C  INPUT :  (I*4)  NDTEM   = MAXIMUM NUMBER OF TEMPERATURES ALLOWED
C  INPUT :  (I*4)  NDLEV   = MAXIMUM NUMBER OF ENERGY LEVELS ALLOWED
C  INPUT :  (I*4)  NDMET   = MAXIMUM NUMBER OF METASTABLE LEVELS ALLOWED
C
C  INPUT :  (I*4)  IT      = INDEX DENOTING THE TEMPERATURE
C  INPUT :  (I*4)  NORD    = NUMBER OF ORDINARY EXCITED LEVELS
C  INPUT :  (I*4)  NMET    = NUMBER OF METASTABLE LEVELS
C
C  INPUT :  (I*4)  IMETR() = INDEX OF METASTABLE IN COMPLETE LEVEL LIST
C                            (ARRAY SIZE = 'NDMET' )
C  INPUT :  (I*4)  IORDR() =INDEX OF ORDINARY EXCITED LEVELS IN COMPLETE
C                            LEVEL LIST.
C                            (ARRAY SIZE = 'NDLEV' )
C
C  INPUT :  (R*8)  CC(,)   = RATE MATRIX COVERING ALL TRANSITIONS
C                            (UNITS: SEC-1)
C                            VALUES FOR GIVEN TEMPERATURE AND DENSITY.
C                            1st DIMENSION: ENERGY LEVEL INDEX
C                            2nd DIMENSION: ENERGY LEVEL INDEX
C  INPUT :  (R*8)  STV()   = RECOMBINATION  CONTRIBUTION  FOR  EACH
C                            NON-METASTABLE/ORDINARY EXCITED LEVELS.
C                            (UNITS: CM**3)
C                            VALUES FOR GIVEN TEMPERATURE AND DENSITY.
C                            DIMENSION: ORDINARY EXCITED LEVEL INDEX
C  INPUT :  (R*8)  VEC(,)  = RECOMBINATION RATE COEFFT. VALUES.
C                            (UNITS: CM**3/SEC-1)
C                            VALUES FOR GIVEN TEMPERATURE AND DENSITY.
C                            1st DIMENSION: TEMPERATURE INDEX ('IT')
C                            2nd DIMENSION: CAPTURING LEVEL INDEX
C
C  OUTPUT:  (R*8)  VRED()  = VECTOR  OF RECOMBINATION RATE CONTRIBUTIONS
C                            FOR EACH METASTABLE LEVEL.
C                            (UNITS: SEC-1)
C                            VALUES FOR GIVEN TEMPERATURE AND DENSITY.
C                            DIMENSION: METASTABLE LEVEL INDEX
C
C           (I*4)  IM      = METASTABLE LEVEL ARRAY INDEX
C           (I*4)  IS      = ORDINARY EXCITED LEVEL INDEX
C
C
C ROUTINES: NONE
```

```
C
C NOTE:
C           VRED(IM)      =    ( the recombination rate for IM )
C                                          +
C                            SUM( (the transistion rate from ordinary
C                                  level IS to IM)  x  (the recombin-
C                                  ation  contribution  for  ordinary
C                                  level IS) )
C
C                            ABOVE SUM IS OVER ALL ORDINARY LEVELS.
C
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 4569
C
C DATE:    09/10/90
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IMETR(NDMET),             IORDR(NDLEV)
      INTEGER             IT,          NDLEV,       NDMET,       NDTEM
      INTEGER             NMET,        NORD
      REAL*8              CC(NDLEV,NDLEV),          STV(NDLEV)
      REAL*8              VEC(NDTEM,NDLEV),         VRED(NDMET)
```

3.119 bxstvm: Subroutine bxstvm from library adas2xx

```
      SUBROUTINE BXSTVM( NDMET  ,
     &                   NMET   ,
     &                   CRMAT  ,
     &                   VRED   ,
     &                   STVM
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: BXSTVM *********************
C
C  PURPOSE: TO  CALCULATE  AND STACK UP IN  'STVM'  THE METASTABLE LEVEL
C           RECOMBINATION  COEFFICIENTS  FOR  A  GIVEN  TEMPERATURE  AND
C           DENSITY.
C
C  CALLING PROGRAM:  ADAS205/ADAS206
C
C  SUBROUTINE:
C
C  INPUT :  (I*4)  NDMET   = MAXIMUM NUMBER OF METASTABLE LEVELS ALLOWED
C
C  INPUT :  (I*4)  NMET    = NUMBER OF METASTABLE LEVELS
C
C  INPUT :  (R*8)  CRMAT(,)= INVERTED  METASTABLE  LEVEL   RATE   MATRIX
C                            COVERING ALL TRANSITIONS BETWEEN METASTABLE
C                            LEVELS EXCEPT THOSE INVOLVING LEVEL 1.
C                            (UNITS: SEC)
C                            VALUES FOR GIVEN TEMPERATURE AND DENSITY.
C                            1st DIMENSION: METASTABLE LEVEL INDEX - 1
C                            2nd DIMENSION: METASTABLE LEVEL INDEX - 1
C
C  INPUT :  (R*8)  VRED()  = VECTOR  OF RECOMBINATION RATE CONTRIBUTIONS
C                            FOR EACH METASTABLE LEVEL.
C                            (UNITS: SEC-1)
C                            VALUES FOR GIVEN TEMPERATURE AND DENSITY.
C                            DIMENSION: METASTABLE LEVEL INDEX
C
C  OUTPUT:  (R*8)  STVM()  = RECOMBINATION  CONTRIBUTION  FOR  EACH
C                            METASTABLE LEVEL. (UNITS: CM**3)
C                            VALUES FOR GIVEN TEMPERATURE AND DENSITY.
C                            (LEVEL 1 IS TAKEN AS ZERO)
C                            DIMENSION: METASTABLE LEVEL INDEX
C
C           (I*4)  IM1     = METASTABLE LEVEL ARRAY INDEX
C           (I*4)  IM2     = METASTABLE LEVEL ARRAY INDEX
C
C
C ROUTINES: NONE
C
C NOTE:
C           STVM(IM1)       SUM( (the transistion rate from IM2 to IM1)
C                                x (the recombination rate contribution
C                                   for metastable level IM2) )
C
C                           (IM1 & IM2 = METASTABLE LEVEL INDEX)
C
C                           ABOVE SUM IS OVER ALL METASTABLE LEVELS
C                           EXCEPT LEVEL ONE.
C
C
```

```
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 4569
C
C DATE:    09/10/90
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             NDMET,       NMET
      REAL*8              CRMAT(NDMET,NDMET),       STVM(NDMET)
      REAL*8              VRED(NDMET)
```

3.120 bxttyp: Subroutine bxttyp from library adas2xx

```
      SUBROUTINE BXTTYP( NDLEV  , NDMET  , NDTRN  , NPLR  , NPLI  ,
     &                   ITRAN  , TCODE  , I1A    , I2A   , AVAL  ,
     &                   ICNTE  , ICNTP  , ICNTR  , ICNTH , ICNTI ,
     &                   ICNTL  , ICNTS  ,
     &                   IETRN  , IPTRN  , IRTRN  , IHTRN , IITRN ,
     &                   ILTRN  , ISTRN  ,
     &                                     IE1A   , IE2A  , AA    ,
     &                                     IP1A   , IP2A  ,
     &                                     IA1A   , IA2A  , AUGA  ,
     &                                     IL1A   , IL2A  , WVLA  ,
     &                                     IS1A   , IS2A  , LSS04A
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: BXTTYP *********************
C
C  PURPOSE:  TO SORT TRANSITION ARRAYS INTO SEVEN TRANSITION/RECOMB TYPES
C
C  CALLING PROGRAM: General
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  NDLEV   = MAXIMUM NUMBER OF LEVELS THAT CAN BE READ
C  INPUT : (I*4)  NDMET   = MAXIMUM NUMBER OF METASTABLES
C  INPUT : (I*4)  NDTRN   = MAXIMUM NUMBER OF TRANS. THAT CAN BE READ
C
C  OUTPUT: (I*4)  NPLR    = NO. OF ACTIVE METASTABLES OF (Z+1) ION
C  OUTPUT: (I*4)  NPLI    = NO. OF ACTIVE METASTABLES OF (Z-1) ION
C
C  INPUT : (I*4)  ITRAN   = INPUT DATA FILE: NUMBER OF TRANSITIONS
C  INPUT : (C*1)  TCODE() = TRANSITION: DATA TYPE POINTER:
C                           ' ' => Electron Impact   Transition
C                           'P' => Proton   Impact   Transition
C                           'H' => Charge   Exchange Recombination
C                           'R' => Free     Electron Recombination
C                           'I' => Electron Impact Ionisation to z
C                           'L' => Satellites from DR Recombination
C                           'S' => Electron Impact Ionisation to z+1
C  INPUT : (I*4)  I1A()   = TRANSITION:
C                             LOWER ENERGY LEVEL INDEX (CASE ' ' & 'P')
C                             PARENT ENERGY LEVEL INDEX(CASE 'H' & 'R')
C                                                      (         & 'L')
C                             FINAL PARENT LEVEL INDEX (CASE 'S')
C  INPUT : (I*4)  I2A()   = TRANSITION:
C                             UPPER ENERGY LEVEL INDEX (CASE ' ' & 'P')
C                             CAPTURING    LEVEL INDEX (CASE 'H' & 'R')
C                                                      (         & 'L')
C                             IONISING     LEVEL INDEX (CASE 'S')
C  INPUT : (R*8)  AVAL()  = TRANSITION:
C                             A-VALUE (SEC-1)          (CASE ' ')
C                             NEUTRAL BEAM ENERGY      (CASE 'H')
C                             AUGER VALUE(SEC-1)       (CASE 'R')
C                             PARENT WAVLENGTH (A)     (CASE 'L')
C                             NOT USED                 (CASE 'P' & 'S')
C
C  OUTPUT: (I*4)  ICNTE   = NUMBER OF ELECTRON IMPACT TRANSITIONS INPUT
C  OUTPUT: (I*4)  ICNTP   = NUMBER OF PROTON IMPACT TRANSITIONS INPUT
C  OUTPUT: (I*4)  ICNTR   = NUMBER OF FREE ELECTRON RECOMBINATIONS INPUT
C  OUTPUT: (I*4)  ICNTH   = NO. OF CHARGE EXCHANGE RECOMBINATIONS INPUT
```

```
C  OUTPUT: (I*4)  ICNTI   = NO. OF IONISATIONS TO Z INPUT
C  OUTPUT: (I*4)  ICNTL   = NO. OF SATELLITE DR RECOMBINATIONS INPUT
C  OUTPUT: (I*4)  ICNTS   = NO. OF IONISATIONS TO Z+1 INPUT
C
C  OUTPUT: (I*4)  IETRN() = ELECTRON IMPACT TRANSITION:
C                           INDEX VALUES IN MAIN TRANSITION ARRAYS WHICH
C                           REPRESENT ELECTRON IMPACT TRANSITIONS.
C  OUTPUT: (I*4)  IPTRN() = PROTON IMPACT TRANSITION:
C                           INDEX VALUES IN MAIN TRANSITION ARRAYS WHICH
C                           REPRESENT PROTON IMPACT TRANSITIONS.
C  OUTPUT: (I*4)  IRTRN() = FREE ELECTRON RECOMBINATION:
C                           INDEX VALUES IN MAIN TRANSITION ARRAYS WHICH
C                           REPRESENT FREE ELECTRON RECOMBINATIONS.
C  OUTPUT: (I*4)  IHTRN() = CHARGE EXCHANGE RECOMBINATION:
C                           INDEX VALUES IN MAIN TRANSITION ARRAYS WHICH
C                           REPRESENT CHARGE EXCHANGE RECOMBINATIONS.
C  OUTPUT: (I*4)  IITRN() = ELECTRON IMPACT IONISATION:
C                           INDEX VALUES IN MAIN TRANSITION ARRAYS WHICH
C                           REPRESENT IONISATIONS FROM LOWER STAGE ION.
C  OUTPUT: (I*4)  ILTRN() = SATELLITE DR RECOMBINATION:
C                           INDEX VALUES IN MAIN TRANSITION ARRAYS WHICH
C                           REPRESENT SATELLITE DR RECOMBINATIONS.
C  OUTPUT: (I*4)  ISTRN() = ELECTRON IMPACT IONISATION:
C                           INDEX VALUES IN MAIN TRANSITION ARRAYS WHICH
C                           REPRESENT IONISATIONS TO UPPER STAGE ION.
C
C  OUTPUT: (I*4)  IE1A()  = ELECTRON IMPACT TRANSITION:
C                            LOWER ENERGY LEVEL INDEX
C  OUTPUT: (I*4)  IE2A()  = ELECTRON IMPACT TRANSITION:
C                            UPPER ENERGY LEVEL INDEX
C  OUTPUT: (R*8)  AA()    = ELECTRON IMPACT TRANSITION: A-VALUE (SEC-1)
C
C
C  OUTPUT: (I*4)  IP1A()  = PROTON IMPACT TRANSITION:
C                            LOWER ENERGY LEVEL INDEX
C  OUTPUT: (I*4)  IP2A()  = PROTON IMPACT TRANSITION:
C                            UPPER ENERGY LEVEL INDEX
C
C  OUTPUT: (I*4)  IA1A()  = AUGER TRANSITION:
C                            PARENT ENERGY LEVEL INDEX
C  OUTPUT: (I*4)  IA2A()  = AUGER TRANSITION:
C                            RECOMBINED ION ENERGY LEVEL INDEX
C  OUTPUT: (R*8)  AUGA()  = AUGER TRANSITION: AUG-VALUE (SEC-1)
C                            RECOMBINED ION ENERGY LEVEL INDEX
C  OUTPUT: (I*4)  IL1A()  = SATELLITE DR TRANSITION:
C                            RECOMBINING ION  INDEX
C  OUTPUT: (I*4)  IL2A()  = SATELLITE DR TRANSITION:
C                            RECOMBINED ION INDEX
C  OUTPUT: (R*8)  WVLA()  = SATELLITE DR TRANSITION: PARENT WVLGTH.(A)
C                            DR SATELLITE LINE INDEX
C  OUTPUT: (I*4)  IS1A()  = IONISING TRANSITION:
C                            IONISED ION  INDEX
C  OUTPUT: (I*4)  IS2A()  = IONISING TRANSITION:
C                            IONISING ION INDEX
C  OUTPUT: (L*4)  LSS04A(,)= .TRUE. => IONIS. RATE SET IN ADF04 FILE:
C                            .FALSE.=> NOT SET IN ADF04 FILE
C                            1ST DIM: LEVEL INDEX
C                            2ND DIM: PARENT METASTABLE INDEX
C
C          (I*4)  I       = GENERAL USE.
C
```

```
C
C ROUTINES: NONE
C
C AUTHOR:  HP SUMMERS (REVISION OF BXTTYP BY PE BRIDEN)
C          K1/1/57
C          JET EXT. 4941
C
C DATE  :  11/06/92
C
C-----------------------------------------------------------------------
C PUT UNDER SCCS CONTROL:
C
C VERSION: 1.1 DATE: 10/05/96
C MODIFIED: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C     - FIRST PUT UNDER SCCS
C
C VERSION: 1.2 DATE: 13/09/99
C MODIFIED: HUGH SUMMERS, UNIVERSITY OF STRATHCLYDE
C     - ADDED DETECTION OF L-LINES AND S-LINES
C
C-----------------------------------------------------------------------
C
C VERSION: 1.2 DATE: 01/05/2003
C MODIFIED: Martin O'Mullane
C     - Replaced original bxttyp with b8ttyp version 1.2.
C Hence the 1.2 version no.
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
C
C VERSION: 1.3 DATE: 17/03/2005
C MODIFIED: Allan Whiteford
C     - Made the routine accept that transition codes of '1',
C             '2' and '3' as well as ' ' correspond to electron
C             impact excitation.
C
C-----------------------------------------------------------------------
      CHARACTER           TCODE(NDTRN)
      INTEGER             I1A(NDTRN),  I2A(NDTRN),  IA1A(NDTRN)
      INTEGER             IA2A(NDTRN), ICNTE,       ICNTH,       ICNTI
      INTEGER             ICNTL,       ICNTP,       ICNTR,       ICNTS
      INTEGER             IE1A(NDTRN), IE2A(NDTRN), IETRN(NDTRN)
      INTEGER             IHTRN(NDTRN),             IITRN(NDTRN)
      INTEGER             IL1A(NDLEV), IL2A(NDLEV), ILTRN(NDTRN)
      INTEGER             IP1A(NDTRN), IP2A(NDTRN), IPTRN(NDTRN)
      INTEGER             IRTRN(NDTRN),             IS1A(NDLEV)
      INTEGER             IS2A(NDLEV), ISTRN(NDTRN),             ITRAN
      INTEGER             NDLEV,       NDMET,       NDTRN,       NPLI
      INTEGER             NPLR
      LOGICAL             LSS04A(NDLEV,NDMET)
      REAL*8              AA(NDTRN),   AUGA(NDTRN), AVAL(NDTRN)
      REAL*8              WVLA(NDLEV)
```

3.121 bxwr11: Subroutine bxwr11 from library adas2xx

```
      SUBROUTINE BXWR11( IUNIT  , DSNINC , TITLED ,
     &                   NDLEV  , NDTEM  , NDDEN  , NDMET ,
     &                   IZ     , IZ0    , IZ1    , BWNO  ,
     &                   IL     , NMET   , NORD   ,
     &                   MAXT   , MAXD   , ICNTR  , ICNTH ,
     &                   IA     , ISA    , ILA    , XJA   ,
     &                   CSTRGA ,
     &                   IMETR  , IORDR  , TEA    , DENSA ,
     &                   STCKM  , STVR   , STVH   ,
     &                   STVRM  , STVHM  , STACK
     &                 )
C-----------------------------------------------------------------------
C
C  ******************** FORTRAN77 SUBROUTINE: BXWR11 ******************
C
C  PURPOSE:  TO OUTPUT DATA TO CONTOUR PASSING FILE.
C            POPULATION  DATA  FOR DIAGNOSTIC USE.
C
C  CALLING PROGRAM: ADAS205/ADAS206
C
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  IUNIT   = OUTPUT UNIT NUMBER FOR RESULTS
CA INPUT : (C*80) DSNINC  = INPUT COPASE DATA SET NAME (IN QUOTES).
C  INPUT : (C*3)  TITLED  = ELEMENT SYMBOL.
C
C  INPUT : (I*4)  NDLEV   = MAXIMUM NUMBER OF LEVELS ALLOWED
C  INPUT : (I*4)  NDTEM   = MAXIMUM NUMBER OF TEMPERATURES ALLOWED
C  INPUT : (I*4)  NDDEN   = MAXIMUM NUMBER OF DENSITIES ALLOWED
C  INPUT : (I*4)  NDMET   = MAXIMUM NUMBER OF METASTABLES ALLOWED
C
C  INPUT : (I*4)  IZ      =  RECOMBINED ION CHARGE READ
C  INPUT : (I*4)  IZ0     =         NUCLEAR CHARGE READ
C  INPUT : (I*4)  IZ1     = RECOMBINING ION CHARGE READ
C                           (NOTE: IZ1 SHOULD EQUAL IZ+1)
C  INPUT : (R*8)  BWNO    = IONISATION POTENTIAL (CM-1)
C
C  INPUT : (I*4)  IL      = NUMBER OF ENERGY LEVELS
C  INPUT : (I*4)  NMET    = NUMBER OF METASTABLES LEVELS: 1<=NMET<=NDMET
C  INPUT : (I*4)  NORD    = NUMBER OF ORDINARY LEVELS ('IL' - 'NMET')
C
C  INPUT : (I*4)  MAXT    = NUMBER OF INPUT TEMPERATURES ( 1 -> 'NDTEM')
C  INPUT : (I*4)  MAXD    = NUMBER OF INPUT DENSITIES ( 1 -> 'NDDEN')
C  INPUT : (I*4)  ICNTR   = NUMBER OF FREE ELECTRON RECOMBINATIONS INPUT
C  INPUT : (I*4)  ICNTH   = NO. OF CHARGE EXCHANGE RECOMBINATIONS INPUT
C
C  INPUT : (I*4)  IA()    = ENERGY LEVEL INDEX NUMBER
C  INPUT : (I*4)  ISA()   = MULTIPLICITY FOR LEVEL 'IA()'
C                           NOTE: (ISA-1)/2 = QUANTUM NUMBER (S)
C  INPUT : (I*4)  ILA()   = QUANTUM NUMBER (L) FOR LEVEL 'IA()'
C  INPUT : (R*8)  XJA()   = QUANTUM NUMBER (J-VALUE) FOR LEVEL 'IA()'
C                           NOTE: (2*XJA)+1 = STATISTICAL WEIGHT
C
C  INPUT : (C*18) CSTRGA()= NOMENCLATURE/CONFIGURATION FOR LEVEL 'IA()'
C
C  INPUT : (I*4)  IMETR() = INDEX OF METASTABLE IN COMPLETE LEVEL LIST
C  INPUT : (I*4)  IORDR() = INDEX OF ORDINARY LEVELS IN COMPLETE LEVEL
C                           LIST.
```

```
C  INPUT : (R*8)  TEA()   = ELECTRON TEMPERATURES (UNITS: KELVIN)
C  INPUT : (R*8)  DENSA() = ELECTRON DENSITIES  (UNITS: CM-3)
C
C  INPUT : (R*8) STCKM(,,) = METASTABLE POPULATIONS STACK
C                             1st DIMENSION: METASTABLE INDEX
C                             2nd DIMENSION: TEMPERATURE INDEX
C                             3rd DIMENSION: DENSITY INDEX
C  INPUT : (R*8) STVR(,,)  = FREE ELECTRON RECOMBINATION COEFFICIENTS
C                             1st DIMENSION: ORDINARY LEVEL INDEX
C                             2nd DIMENSION: TEMPERATURE INDEX
C                             3rd DIMENSION: DENSITY INDEX
C  INPUT : (R*8) STVH(,,)  =  CHARGE EXCHANGE COEFFICIENTS
C                             1st DIMENSION: ORDINARY LEVEL INDEX
C                             2nd DIMENSION: TEMPERATURE INDEX
C                             3rd DIMENSION: DENSITY INDEX
C  INPUT : (R*8) STVRM(,,) = METASTABLE FREE ELECTRON RECOMBINATION
C                            COEFFICIENTS.
C                             1st DIMENSION: METASTABLE INDEX
C                             2nd DIMENSION: TEMPERATURE INDEX
C                             3rd DIMENSION: DENSITY INDEX
C  INPUT : (R*8) STVHM(,,) = METASTABLE CHARGE EXCHANGE COEFFICIENTS
C                             1st DIMENSION: METASTABLE INDEX
C                             2nd DIMENSION: TEMPERATURE INDEX
C                             3rd DIMENSION: DENSITY INDEX
C  INPUT : (R*4) STACK(,,,)= POPULATION DEPENDENCE
C                             1st DIMENSION: ORDINARY LEVEL INDEX
C                             2nd DIMENSION: METASTABLE INDEX
C                             3rd DIMENSION: TEMPERATURE INDEX
C                             4th DIMENSION: DENSITY INDEX
C
C          (I*4) I         = GENERAL USE
C          (I*4) J         = GENERAL USE
C          (I*4) K         = GENERAL USE
C          (I*4) L         = GENERAL USE
C
C NOTE:
C          THIS OUTPUT DATA IS FOR SUBSEQUENT INPUT INTO THE DIAGNOSTIC
C          AND CONTOUR GRAPHING PROGRAM 'CONTOUR'.
C
C
C ROUTINES: NONE
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/37
C          JET EXT. 5023
C
C DATE:    09/10/90
C
C
C UPDATE:  20/05/93 - PE BRIDEN - ADAS91: TO REFLECT CHANGES IN BXDATA
C                                         THE FOLLOWING ARRAY DIMENSION/
C                                         SIZE CHANGES WERE MADE:
C                                       1) CHARACTER CSTRGA *12 -> *18
C                                          (FORMAT STMT 1003 CHANGED)
C
C UPDATE:  20/05/93- P BRIDEN: STACK ARRAY CHANGED FROM REAL*8 -> REAL*4
C
C-----------------------------------------------------------------------
      CHARACTER*18        CSTRGA(NDLEV)
      CHARACTER*80        DSNINC
      CHARACTER*3         TITLED
```

```
INTEGER             IA(NDLEV),    ICNTH,        ICNTR,        IL
INTEGER             ILA(NDLEV),   IMETR(NDMET)
INTEGER             IORDR(NDLEV),               ISA(NDLEV),   IUNIT
INTEGER             IZ,           IZ0,          IZ1,          MAXD
INTEGER             MAXT,         NDDEN,        NDLEV,        NDMET
INTEGER             NDTEM,        NMET,         NORD
REAL*8              BWNO,         DENSA(NDDEN)
REAL                STACK(NDLEV,NDMET,NDTEM,NDDEN)
REAL*8              STCKM(NDMET,NDTEM,NDDEN)
REAL*8              STVH(NDLEV,NDTEM,NDDEN)
REAL*8              STVHM(NDMET,NDTEM,NDDEN)
REAL*8              STVR(NDLEV,NDTEM,NDDEN)
REAL*8              STVRM(NDMET,NDTEM,NDDEN), TEA(NDTEM)
REAL*8              XJA(NDLEV)
```

3.122 chindx: Subroutine chindx from library adas2xx

```
       SUBROUTINE CHINDX(  CNJL   ,  INDJL   ,  NJLEVX  ,  NJLEN  ,
     &                     CNBL   ,  INDBL   ,  BNDLS   ,  NGAP   ,
     &                     INDBLO
     &                   )
C-----------------------------------------------------------------------
C  FORTRAN 77 SUBROUTINE CHINDX
C
C  PURPOSE: TO RE-INDEX & ALTER TERMS FOR A BUNDLED SET OF LEVELS
C
C  AUTHOR: DAVID H.BROOKS
C
C  DATE: 28.04.95
C
C  MODIFIED: 23.01.96 DAVID H.BROOKS
C            SLIGHT ALTERATION TO METHOD TO TRAP SOME WIDER CASES.
C
C  VERSION: 1.3 DATE: 18.11.98
C  MODIFIED: DAVID H.BROOKS
C            FURTHER ALTERATION TO TRAP SOME WIDER CASES.
C-----------------------------------------------------------------------
      CHARACTER*18        CNBL(NJLEVX),             CNJL(NJLEVX)
      INTEGER             BNDLS(NJLEVX),            INDBL(NJLEVX)
      INTEGER             INDBLO(NJLEVX),           INDJL(NJLEVX)
      INTEGER             NGAP(NJLEVX),             NJLEN,       NJLEVX
```

3.123 dielcl: Subroutine dielcl from library adas2xx

```
       SUBROUTINE DIELCL(Z,EIJ,F,T,COR,JCOR,N,DEF,AD,L,CPT)
       IMPLICIT REAL*8 (A-H,O-Z)
C-----------------------------------------------------------------------
C
C  ************** FORTRAN77 SUBROUTINE: DIELCL  ************************
C
C  VERSION:  1.0
C
C  PURPOSE:  THIS SUBROUTINE IS NOT YET PROPERLY DOCUMENTED
C
C-----------------------------------------------------------------------
C
C UNIX-IDL CONVERSION:
C
C VERSION: 1.1                          DATE: 22-08-96
C MODIFIED: WILLIAM OSBORN
C               - FIRST CONVERTED. NO CHANGES.
C
C VERSION: 1.2                          DATE: 09-09-96
C MODIFIED: WILLIAM OSBORN
C               - COMMENTED-OUT LINE 'CPT=CPTS(L+1)' WHICH WAS GIVING
C                 A COMPILER WARNING AND IS NOT YET NEEDED.
C-----------------------------------------------------------------------

      INTEGER             JCOR,        L,           N
      REAL*8              AD,          COR(20),     CPT,         DEF
      REAL*8              EIJ,         F,           T,           Z
```

3.124 dnaq: Subroutine dnaq from library adas2xx

```
      SUBROUTINE DNAQ(A0,A,B0,B,Q,NMAX,JSWICH)
      IMPLICIT REAL*8(A-H,O-Z)
C-----------------------------------------------------------------------
C      PURPOSE: GIVEN A0 AND ITS FIRST NMAX DERIVATIVES IN ARRAY A,
C      AND GIVEN Q AND NMAX, CALCULATES B0 AND ARRAY B, BEING THE
C      VALUE AND FIRST NMAX DERIVATIVES OF (A0)**Q
C
C      FOR PERHAPS GREATER SPEED, YOU MAY SET JSWICH TO 2 IF Q IS -2.0
C      JSWICH TO 3 IF Q IS -1.0
C      JSWICH TO 4 IF Q IS -0.5
C      JSWICH TO 5 IF Q IS -0.25
C      JSWICH TO 6 IF Q IS 0.25
C      JSWICH TO 7 IF Q IS 0.5
C      JSWICH TO 8 IF Q IS 2.0
C      OTHERWISE SET JSWICH TO 1
C-----------------------------------------------------------------------
C UNIX-IDL PORT:
C
C AUTHOR:  WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C
C DATE:    4TH JULY 1996
C
C VERSION: 1.1                          DATE: 04-07-96
C MODIFIED: WILLIAM OSBORN
C               - FIRST VERSION.
C
C VERSION: 1.2                          DATE: 19-12-01
C MODIFIED: Martin O'Mullane
C               - Removed junk from > column 72.
C
C VERSION: 1.3                          DATE: 16-05-07
C MODIFIED: Allan Whiteford
C               - Modified comments as part of subroutine documentation
C                 procedure.
C
C VERSION: 1.4                          DATE: 20-07-07
C MODIFIED: Allan Whiteford
C               - Further modification to comments as part of
C                 subroutine documentation procedure.
C
C-----------------------------------------------------------------------
      INTEGER             JSWICH,      NMAX
      REAL*8              A(20),       A0,          B(20),       B0
      REAL*8              Q
```

3.125 effz3: Subroutine effz3 from library adas2xx

```
       subroutine effz3( jealfa , n     , l      , e     , qd    ,
     &                   jsn    , z0    , nshell , nc    , numel ,
     &                   alfa   , jalf1 , jalf2  ,
     &                   x0     , x1    , x2     , d     , m0
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** fortran77 program: effz3.for *********************
C
C  Purpose: Searches for the effective potential for a single electron
C           distorted wave function for a specified screening or a
C           specified energy.
C           (original by A. Burgess, DAMTP, University of Cambridge)
C
C
C  Subroutine:
C
C  input : (i*4)  jealfa   = <0 => search for energy e
C                          = >0 => search for screening parameter alfa
C  input : (i*4)  n        = principal quantum number
C  input : (i*4)  l        = orbital quantum number
C  i/o   : (i*4)  e        = energy (Ryd) for electron.
C                            (NB -ve for a bound state)
C  input : (i*4)  qd       = quantum defect for valence electron
C  input : (i*4)  jsn      = -1 => Jucys potential form adopted
C                          = 0  => Slater potential form adopted
C  input : (i*4)  z0       = nuclear charge
C  input : (i*4)  nshell   = number of screening shells
C  input : (i*4)  nc()     = principal quantum number of screening shell
C                            1st dim: index of screening shells
C  input : (i*4)  numel()  = number of electrons in screening shell
C  i/o   : (r*8)  alfa()   = screening parameters
C                            1st dim: initial (1) and final (2) states
C                            2nd dim: screening shell index.
C  input : (i*4)  jalf1    = first screening shell for optimising
C  input : (i*4)  jalf2    = last screening shell for optimising
C  output: (i*4)  x0       = inner turning point
C  output: (i*4)  x1       = outer turning point
C  output: (i*4)  x2       = range for active electron wave function
C  input : (i*4)  d        = earch accuracy setting
C  output: (i*4)  m0       = number of nodes in wave function
C
C
C
C  Routines:
C          routine    source    brief description
C          ------------------------------------------------------------
C          zeff       adas
C          zser       adas
C          fcf6       adas
C          i4unit     adas      fetch unit number for output of messages
C
C
C  Author:  H. P. Summers, University of Strathclyde
C           ja7.08
C           tel. 0141-548-4196
C
C  Date:   24/02/03
C
```

```
C  Update: HP Summers  24/05/04  restructure and addded standard warning
C
C-----------------------------------------------------------------------
      INTEGER             JALF1,       JALF2,       JEALFA,      JSN
      INTEGER             L,           M0,          N,           NC(10)
      INTEGER             NSHELL,      NUMEL(10)
      REAL*8              ALFA(10),    D,           E,           QD
      REAL*8              X0,          X1,          X2,          Z0
```

3.126 escape: Subroutine escape from library adas2xx

```
       subroutine escape( ndprof, nddens, ndcyl , iz    ,
     &                    tg    , j0    , mm    , x0    , x1    ,
     &                    y0    , sum1  , sum2  , sum3  ,
     &                    sum4  , alpha   , ics , iden    ,
     &                    sum   , iprofile , zlen, y, x, wid,
     &   dmult, modprof)
c
c-----------------------------------------------------------------------------
c
c  *********&********** fortran77 subroutine: escape *************************
c
c  original name: escape.bas
c                 (developed by K. H. Behringer)
c
c  version:  1.0
c
c  purpose: computes escape factors.
c
c  calling program: adas214
c
c  input:
c          (i*4)  nddens  = parameter = max. number of points over ion profile
c          (i*4)  ndcyl   = parameter = max. number of intervals for cylinder
c          (i*4)  ndprof  = parameter = max. number of points over line profile
c
c    (i*4)  ics     = 1 for slab geometry of plasma
c                   = 2 for cyl. geometry of plasma
c          (i*4)  iden    = 1 for homogenous density distribution
c                         = 2 for linear density
c                         = 3 for parabolic density
c          (i*4)  iprofile= 1 for Doppler line profile
c                         = 2 for Lorentzian line profile
c                         = 3 for Holtzmark line profile
c                         = 10 for Double Doppler line profile
c                         = -2 for Voigt line profile
c                         = -3 for Doppler Holtzmark line profile
c          (i*4)  iz      = number of points over line profile
c          (i*4)  j0      = number of density values for integration over
c                           cylinder/slab
c          (i*4)  k0      = number of density values - must be enough for
c                           interpolation
c          (i*4)  z0      = number of intervals for cylinder/slab
c
c          (r*8)  allam   = Spectral absorption coefficient
c          (r*8)  alpha   = neutral density*oscillator strength*length
c          (r*8)  domeg   = delta omega in cylinder/slab
c          (r*8)  l       = Length of plasma (0 to inf. on one side)
c          (r*8)  mm      = atomic mass number
c          (r*8)  omega   = solid angle in cylinder/slab
c          (r*8)  r       = radius of cylinder
c          (r*8)  sum1()  = sum assuming homogeneous density
c          (r*8)  sum2()  = sum assuming linear density
c          (r*8)  sum3()  = sum assuming parabolic density
c          (r*8)  sum4()  = used in sum for escape factor
c          (r*8)  sum()   = sum1/2/3, depending upon options chosen for density profile
c          (r*8)  tg      = neutral temperature  (eV)
c          (r*8)  w       = Full Doppler width calculated for lambda=100nm
c          (r*8)  x0()    = Absorption coefficient at line centre/10
c          (r*8)  x1()    = Absorption coefficient at line centre
```

```
c          (r*8)  y       = Line profile intensity
c          (r*8)  y0      = Line profile intensity (?) at line centre
c          (r*8)  ff      = Used in calculation of escape factor
c          (r*8)  xl      = log of x1, used for interpolation
c          (r*8)  yl      = log of sum, used in interpolation
c          (r*8)  zlen    = variable used to select length along plasma integration to be
c
c
c  routines:
c           routine     source     brief description
c           ------------------------------------------------------------------
c           faltung     ADAS       computes line profiles
c
c  author:  K. H. Behringer  (IPF, University of Stuttgart)
c
c  date:    30/01/94
c
c  update:  ???? brought up to date with latest Behringer code
c
c  VERSION:  1.1 DATE: 18-06-98
c  MODIFIED: STUART LOCH
c - CONVERTED TO FORTRAN
c
c  VERSION:  1.2 DATE: 26-11-98
c  MODIFIED: STUART LOCH
c - CHECKS IF LINE PROFILE IS INTEGRATED FAR ENOUGH,
c   IF NOT THE PROFILE IS RE-EVALUATED TO TWICE THE
c   PREVIOUS WIDTH, USING THE SAME STEP SIZE AS BEFORE.
c  VERSION: 1.3                                          DATE: 19-02-99
c  MODIFIED: STUART lOCH
c               -DIMENSIONS OF X,Y, YT, YL AND MODPROF INCREASED FROM 1000
c                TO 3000, TO ACCOMODATE POSSIBLE INCREASES IN INTEGRATION
c                LIMIT OF LINE PROFILE
c  VERSION: 1.4                                          DATE: 24-09-99
c  MODIFIED: STUART lOCH
c               -REMOVED EMPTY DO LOOP
C
c-----------------------------------------------------------------------------
      INTEGER             ICS,          IDEN,         IPROFILE,     IZ
      INTEGER             J0,           NDCYL,        NDDENS,       NDPROF
      REAL*8              ALPHA,        DMULT,        MM
      REAL*8              MODPROF(K0,3000),           SUM(NDDENS)
      REAL*8              SUM1(NDDENS),               SUM2(NDDENS)
      REAL*8              SUM3(NDDENS),               SUM4(NDDENS)
      REAL*8              TG,           WID,          X(3000)
      REAL*8              X0(NDDENS),   X1(NDDENS),   Y(3000),      Y0
      REAL*8              ZLEN
```

3.127 fintb: Subroutine fintb from library adas2xx

```
       REAL*8 FUNCTION FINTB(X)
C-----------------------------------------------------------------------
C
C  ************** FORTRAN77 SUBROUTINE: FINTB   ************************
C
C  VERSION:  1.0
C
C  PURPOSE:  UNKNOWN
C
C  THIS SUBROUTINE IS NOT YET PROPERLY DOCUMENTED
C
C-----------------------------------------------------------------------
C
C UNIX-IDL CONVERSION:
C
C VERSION: 1.1                          DATE: 22-08-96
C MODIFIED: WILLIAM OSBORN
C               - FIRST CONVERTED. NO CHANGES.
C
C VERSION: 1.2                          DATE: 20-09-99
C MODIFIED:     RICHARD MARTIN
C               ADDED "REAL*8" TO  "FUNCTION FINTB(X)"
C
C-----------------------------------------------------------------------
       IMPLICIT REAL*8 (A-H,O-Z)
       FINTB=X
       RETURN
       END
      REAL*8              X
```

3.128 gext: Subroutine gext from library adas2xx

```
       FUNCTION GEXT(X,N,L)
       IMPLICIT REAL*8(A-H,O-Z)
C
C  PURPOSE: PRODUCES ONE ELECTRON ORBITALS FROM SPECIFIED
C           FUNCTIONAL FORMS
C
C  FOR USE IN DWBES, RDWBES,DWDIP WITH EXTERNAL OPTION IEXT=1
C  ____________________________________________________________________
C  HIBBERT (CIV3 PROGRAM) ORBITALS FOR OII 24/4/85
C  ____________________________________________________________________
C  INDEXING OF WAVE FUNCTIONS BY I=(N*(N-1))/2+L+1
C UNIX-IDL PORT:
C
C----------------------------------------------------------------------
C AUTHOR:  WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C
C DATE:    4TH JULY 1996
C
C VERSION: 1.1                          DATE: 04-07-96
C MODIFIED: WILLIAM OSBORN
C               - FIRST VERSION.
C
C VERSION: 1.2                          DATE: 20-07-07
C MODIFIED: Allan Whiteford
C               - Small modification to comments to allow for
C                 automatic documentation preparation.
C----------------------------------------------------------------------
C
       I=(N*(N-1))/2+L+1
       GO TO (1,2,3,4,5,6),I
    1  GEXT=X*(38.1978304D0*DEXP(-7.4780300*X)+4.9817906D0*DEXP(-12.6307
     &000D0*X))+X*X*(0.0928714D0*DEXP(-3.1009000D0*X)+2.1368144D0*DEXP(-
     &6.3727700D0*X)-0.0087901D0*DEXP(-2.0732300D0*X))
       RETURN
    2  GEXT=X*(-9.6934267D0*DEXP(-7.4780300D0*X)-0.5036558D0*DEXP(-12.63
     &07000D0*X))+X*X*(9.2494101D0*DEXP(-3.1009000D0*X)-11.2771775D0*DEX
     &P(-6.3727700D0*X)+4.5358980D0*DEXP(-2.0732300D0*X))
       RETURN
    3  GEXT=X*X*(4.5603425D0*DEXP(-2.2378000D0*X)+7.9197229D0*DEXP(-3.82
     &44700D0*X)+1.1697093D0*DEXP(-1.6770200D0*X)+2.6575680D0*DEXP(-8.58
     &10500D0*X))
       RETURN
    4  GEXT=X*(3.7467103D0*DEXP(-6.4449722D0*X))+X*X*(-5.0254465D0*DEXP(
     &-2.4960885D0*X))+X*X*X*(0.5162317D0*DEXP(-1.0396983D0*X))
       RETURN
    5  GEXT=X*X*(4.0687494D0*DEXP(-2.6750348D0*X))+X*X*X*(-0.2285374D0*D
     &EXP(-0.8361112D0*X))
       RETURN
    6  GEXT=X*X*X*(0.1289291D0*DEXP(-0.7128119D0*X))
       RETURN
      END
      INTEGER             L,           N
      REAL*8              X
```

3.129 isort: Subroutine isort from library adas2xx

```
       SUBROUTINE ISORT(ND,IAIN,IAOUT,IVSORT,N,LPAPER)
C-----------------------------------------------------------------------
C  PURPOSE: SORT AN INTEGER ARRAY SO THAT IT IS IN INCREASING ORDER
C
C  INPUT
C      ND       = MAXIMUM DIMENSION OF ARRAY
C      IAIN(I)  = INPUT ARRAY
C      N        = NUMBER OF VALUES
C  OUTPUT
C      IAOUT(I) = SORTED ARRAY VALUES
C      IVSORT(I)= INVERSE SORT VALUES
C
C
C AUTHOR: UNKOWN
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 31-7-96
C MODIFIED: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST VERSION
C
C VERSION: 1.2                          DATE: 20-7-07
C MODIFIED: Allan Whiteford
C               - Small modification to comments to allow for
C                 automatic documentation preparation.
C               - Removed junk from columnds > 72.
C               - Removed old SCCS header.
C
C-----------------------------------------------------------------------
      INTEGER             IAIN(ND),    IAOUT(ND),   IVSORT(ND),  N
      INTEGER             ND
      LOGICAL             LPAPER
```

3.130 ngffmh: Subroutine ngffmh from library adas2xx

```
       REAL*8 FUNCTION NGFFMH(GAM2)
       IMPLICIT REAL*8(A-H,O-Z)
C-----------------------------------------------------------------------
C
C  ************** FORTRAN77 SUBROUTINE: NGFFMH  ************************
C
C  VERSION:  1.0
C
C  PURPOSE: EVALUATES ELECTRON TEMPERATURE AND FREQUENCY AVERAGED
C           HYDROGENIC FREE FREE GAUNT FACTOR.
C
C  OBTAINED FROM INTERPOLATION OF KARZAS & LATTER (1959) FIG.6
C  FOR -3<LOG10(Z0*Z0*IH/KTE)<1. OUTSIDE THIS RANGE A VERY APPROXIMATE
C  EXTRAPOLATION IS PERFORMED WITH GFFMH=1 IN THE INFINITE LIMITS.
C
C  INPUT:
C       GAM2=Z0*Z0*IH/KTE
C  OUTPUT:
C       NGFFMH=MAXWELL AND FREQUENCY AVERAGED FREE-FREE GAUNT FACTOR.
C  ********* H.P.SUMMERS, JET         12 JAN 1987    ******************
C-----------------------------------------------------------------------
C
C UNIX-IDL CONVERSION:
C
C VERSION: 1.1                          DATE: 22-08-96
C MODIFIED: WILLIAM OSBORN
C               - FIRST CONVERTED. NO CHANGES.
C
C VERSION: 1.2                          DATE: 13-09-99
C MODIFIED: Martin O'Mullane
C               - Define function name as real*8.
C
C VERSION: 1.3                          DATE: 16-05-07
C MODIFIED: Martin O'Mullane
C               - Modified comments as part of subroutine documentation
C                 procedure.
C
C-----------------------------------------------------------------------
      REAL*8              GAM2
```

3.131 pchg: Subroutine pchg from library adas2xx

```
       FUNCTION PCHG(V,L,L1,E,FACT)
       IMPLICIT REAL*8(A-H,O-Z)
C-----------------------------------------------------------------------
C
C  PURPOSE: EVALUATES PEACH AMPLITUDE G(V,L,E,L1) BY TABLE INTERPOLATION
C
C  TABLES ARE READ IN FROM DISC FILE 'JETSHP.PCHGTAB.DATA' ON FIRST
C  CALL TO PCHG AND STORED IN LABELLED COMMON /PCHGTB/.
C  THE INPUT IS ON STREAM 13.  IGONE MUST BE SET TO 1 IN MAIN ROUTINE
C  INITIALLY. IT IS RESET TO 0 AFTER TABLES ARE READ IN.
C  ********** H.P. SUMMERS, JET     26 JUNE 1985  ********************
C_______________________________________________________________________
C  INPUT
C      N=EFFECTIVE PRINCIPAL QUANTUM NUMBER OF BOUND STATE
C      L=ORBITAL ANGULAR MOMENTUM OF ELECTRON IN BOUND STATE
C      L1=ORBITAL ANGULAR MOMENTUM OF ELECTRON IN FREE STATE
C      E=ELECTRON ENERGY IN FREE STATE (REDUCED RYDBERG UNITS)
C  OUTPUT
C      PCHG=PEACH INTERPOLATED AMPLITUDE G OR G*
C      FACT=1                IF PEACH G  RETURNED
C          =DSQRT(DABS(V-L)) IF PEACH G* RETURNED
C                            NB OBTAIN G FROM G* BY MULTIPLYING BY FACT
C_______________________________________________________________________
C-----------------------------------------------------------------------
C UNIX-IDL PORT:
C
C AUTHOR:  WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C
C DATE:    4TH JULY 1996
C
C VERSION: 1.1                          DATE: 04-07-96
C MODIFIED: WILLIAM OSBORN
C               - FIRST VERSION.
C
C VERSION: 1.2                          DATE: 19-12-01
C MODIFIED: Martin O'MULLANE
C               - Removed junk from > column 72.
C
C VERSION: 1.3                          DATE: 20-07-07
C MODIFIED: Allan Whiteford
C               - Small modification to comments to allow for
C                 automatic documentation preparation.
C
C-----------------------------------------------------------------------
      INTEGER             L,           L1
      REAL*8              E,           FACT,        V
```

3.132 pchx: Subroutine pchx from library adas2xx

```
       FUNCTION PCHX(V,L,L1,E)
       IMPLICIT REAL*8(A-H,O-Z)
C-----------------------------------------------------------------------
C
C  PURPOSE: EVALUATES PEACH PHASE X(V,L,E,L1) BY TABLE INTERPOLATION
C
C  TABLES ARE READ IN FROM DISC FILE 'JETSHP.PCHXTAB.DATA' ON FIRST
C  CALL TO PCHX AND STORED IN LABELLED COMMON /PCHXTB/.
C  THE INPUT IS ON STREAM 14.  IFIRST MUST BE SET TO 1 IN MAIN ROUTINE
C  INITIALLY. IT IS RESET TO 0 AFTER TABLES ARE READ IN.
C  ********** H.P. SUMMERS, JET     26 JUNE 1985  ********************
C_______________________________________________________________________
C  INPUT
C      N=EFFECTIVE PRINCIPAL QUANTUM NUMBER OF BOUND STATE
C      L=ORBITAL ANGULAR MOMENTUM OF ELECTRON IN BOUND STATE
C      L1=ORBITAL ANGULAR MOMENTUM OF ELECTRON IN FREE STATE
C      E=ELECTRON ENERGY IN FREE STATE (REDUCED RYDBERG UNITS)
C  OUTPUT
C      PCHX=PEACH INTERPOLATED PHASE
C_______________________________________________________________________
C-----------------------------------------------------------------------
C UNIX-IDL PORT:
C
C AUTHOR:  WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C
C DATE:    4TH JULY 1996
C
C VERSION: 1.1                          DATE: 04-07-96
C MODIFIED: WILLIAM OSBORN
C               - FIRST VERSION.
C
C VERSION: 1.2                          DATE: 19-12-01
C MODIFIED: Martin O'MULLANE
C               - Removed junk from > column 72.
C
C VERSION: 1.3                          DATE: 20-07-07
C MODIFIED: Allan Whiteford
C               - Small modification to comments to allow for
C                 automatic documentation preparation.
C
C-----------------------------------------------------------------------
      INTEGER             L,          L1
      REAL*8              E,          V
```

3.133 phase: Subroutine phase from library adas2xx

```
      REAL*8 FUNCTION PHASE(E,EL,Z,X)
C-----------------------------------------------------------------------
C
C PURPOSE: CALCULATES ASYMPTOTIC PHASE OF FREE COULOMB
C          REAL FUNCTION FCF4
C
C-----------------------------------------------------------------------
C UNIX-IDL PORT:
C
C AUTHOR:  WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C
C DATE:    4TH JULY 1996
C
C VERSION: 1.1                          DATE: 04-07-96
C MODIFIED: WILLIAM OSBORN
C               - FIRST VERSION.
C
C VERSION  : 1.2
C DATE     : 19-12-2001
C MODIFIED : Martin O'Mullane
C               - Changed function defintion to a more standard form.
C               - Removed mainframe listing information beyond column 72.
C
C VERSION  : 1.3
C DATE     : 20-07-2007
C MODIFIED : Allan Whiteford
C               - Small modification to comments to allow for automatic
C                 documentation preparation.
C
C-----------------------------------------------------------------------
      IMPLICIT REAL*8(A-H,O-Z)
      L=EL+0.5
      PI=3.141592654
      ZZ=Z*Z
      CK=DSQRT(E)
       XK=X*CK
      XZ=X*Z
      C=EL*(EL+1.0)
      CHI=DSQRT(XK*XK-XZ-XZ-C)
      C1=1.0/CHI
      IF(C) 1,1,2
    1 THETA=0.25/(CHI+XK)
      GO TO 8
    2 T=CK*C*CHI+ZZ*X+C*Z
      T1=DABS (T)
      IF(T1-1.0D-15)3,3,4
    3 THETA =0.5*PI
      GO TO 7
    4 ARG=DSQRT(C*(ZZ+E*C)*(CK*C*CHI+E*C*X+2.*ZZ*X+C*Z)/(CK*CHI+E*X-
     1Z))/T
      IF(T)5,5,6
    5 THETA =PI-DATAN (-ARG)
      GOTO 7
    6 THETA=DATAN (ARG)
    7  THETA=THETA*(C+0.125)/DSQRT(C)
    8 IF(E-1.0D-50) 9,9,10
    9 PHASE=CHI+CHI+THETA-(EL+0.25)*PI-0.166666667*C1*(1.0+1.25*(XZ+C)*C
     11*C1)
      GO TO 11
```

```
10 A=Z/CK
   B=E*C+ZZ
   D=3.0*(CHI+XK)*B+ZZ*(CHI-XK)-CK*C*Z
   D=(D/(24.0*B*(CHI+XK))+0.208333333*(XZ+C)*C1*C1)*C1
   AG=ARGAM(L,A)
   PHASE=CHI-A*DLOG(CHI+XK-A)+AG+A-0.5*PI*EL+THETA-D
11 RETURN
   END
   REAL*8              E,           EL,          X,           Z
```

3.134 qpoly: Subroutine qpoly from library adas2xx

```
       subroutine qpoly( x1, y1, x2, y2, x3, y3, x, y)
c
c-----------------------------------------------------------------------
C  **************FORTRAN77 SUBROUTINE:QPOLY ************
C VERSION 1.0
C
C PURPOSE :QUADRATIC INTERPOLATION OF X1,Y1,X2,Y2,X3,Y3 -> Y FOR GIVEN X
C
C INPUT :
C  (R*8) X1 =
C  (R*8) Y1 =
C  (R*8) X2 =
C  (R*8) Y2 =
C  (R*8) X3 =
C  (R*8) Y3 =
C
C ROUTINES:
C
C AUTHOR: K.H. BEHRINGER (IPF, UNIVERSITY OF STUTTGART)
C
C DATE 02/04/98
C
C VERSION: 1.1 DATE: 18-06-98
C MODIFIED: STUART LOCH
C - CONVERTED TO FORTRAN FOR ADAS.
C
c-----------------------------------------------------------------------
      REAL*8              X,           X1,          X2,          X3
      REAL*8              Y,           Y1,          Y2,          Y3
```

3.135 r2photo: Subroutine r2photo from library adas2xx

```
       SUBROUTINE R2PHOTO(ATE,EN,RREC,QRREC)
       IMPLICIT REAL*8(A-H,O-Z)
C-----------------------------------------------------------------------
C  *********  FORTRAN 77 SUBROUTINE ********************
C
C            NAME:  R2PHOTO
C
C         VERSION:  2.0
C
C  PREVIOUS NAMES:  JETSHP.NMAINCH(RPHOTO)  (H.P. SUMMERS)
C
C          AUTHOR:  H.P. SUMMERS/ W.J. DICKSON
C
C            DATE:  19/10/93
C
C    PURPOSE:
C    --------
C    UPDATED VERSION OF RPHOTO TO ALLOW
C
C      (I)  RETURN OF ENERGY AVERAGED ELECTRON COOLING COEFFICIENT
C
C
C    INPUT:
C    ------
C    ATE         R*8      EQUAL TO  (157890.0 /TE) *Z*Z
C    EN          R*8      N SHELL
C
C
C    OUTPUT:
C    -------
C    RREC        R*8      ENERGY AVERAGED BOUND-FREE GAUNT FACTOR
C                          (= INT FROM 0 TO INFINITY OF GII*EXP(-X) DX)
C    QRREC       R*8      ENERGY AVERAGED BOUND-FREE GAUNT FACTOR FOR
C                         CALCULATION OF ELECTRON COOLING FUNCTION.
C                          ( = RREC REDUCED BY FACTOR OF E/HV,
C                                WHERE E  = ELECTRON ENERGY
C                                      HV = PHOTON ENERGY )
C
C   NOTES:
C   ------
C    TE IS IN KELVIN UNITS
C    X = E/KT
C
C-----------------------------------------------------------------------
C
C UNIX-IDL CONVERSION:
C
C VERSION: 1.1                          DATE: 22-08-96
C MODIFIED: WILLIAM OSBORN
C               - FIRST CONVERTED. NO CHANGES.
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
C
      REAL*8              ATE,         EN,          QRREC,       RREC
```

3.136 r8necip: Subroutine r8necip from library adas2xx

```
C
      FUNCTION R8NECIP( IZ , XI , ZETA , TE , ALFRED)
      IMPLICIT REAL*8(A-H,O-Z)
C-----------------------------------------------------------------------
C
C  *************** FORTRAN77 REAL*8 FUNCTION: R8NECIP *****************
C
C  PURPOSE: EVALUATES A SHELL CONTRIBUTION TO THE IONISATION RATE
C           COEFFICIENT IN THE ECIP APPROXIMATION, AND RETURNS
C           THE THREE-BODY RECOMBINATION COEFFICIENT IN REDUCED
C           FORM IE. OMITTING THE WEIGHTING FACTOR W(I)/W+
C
C  CF. SUMMERS (1974)APPLETON LABORATORY REPORT IM367
C
C
C  BASED UPON ADASXX20(R8NBCH) AND COPASE(RECIP2)
C
C  CALLING PROGRAM: GENERAL USE
C
C  FUNCTION:
C          (R*8)  R8NECIP = FUNCTION NAME
C          (I*4)  IZ      = TARGET ION CHARGE NUMBER
C                           (RECOMBINED ION CHARGE).
C          (R*8)  XI      = EFFECTIVE IONISATION POTENTIAL FOR SHELL
C                           (UNITS: RYDBERGS)
C                           (LEVEL ENERGY RELATIVE TO IONISATION POT.)
C          (R*8)  ZETA    = EFFECTIVE NUMBER OF EQUIVALENT ELECTRONS
C                           IN SHELL
C          (R*8)  TE      = ELECTRON TEMPERATURE (IN KELVIN)
C          (R*8)  ALFRED  = THREE-BODY COEFFICIENT*W+/W(I)  (CM**6 S-1)
C
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          R8YIP      ADAS      FUNCTION:
C
C
C AUTHOR:   W.J. DICKSON
C           K1/1/36
C           JET EXT. 5057
C
C DATE:     05/01/93
C
C**********************************************************************
C UNIX-IDL PORT:
C
C AUTHOR: DAVID H BROOKS, UNIVERSITY OF STRATHCLYDE
C
C DATE: UNKNOWN
C
C**********************************************************************
C PUT UNDER SCCS CONTROL:
C
C VERSION: 1.1                          DATE: 10/05/96
C MODIFIED: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C           - FIRST PUT UNDER SCCS
C
C VERSION  : 1.2
```

```
C DATE     : 16-02-2006
C MODIFIED : Martin O'Mullane
C               - Removed mainframe listing information beyond column 72.
C
C VERSION  : 1.3
C DATE     : 20-07-2007
C MODIFIED : Allan Whiteford
C               - Small modification to comments to allow for automatic
C                 documentation preparation.
C-----------------------------------------------------------------------
      INTEGER             IZ
      REAL*8              ALFRED,      TE,          XI,          ZETA
```

3.137 rd2fs: Subroutine rd2fs from library adas2xx

```
       FUNCTION RD2FS(N,L,L2,E2)
C-----------------------------------------------------------------------
C  PURPOSE: GENERATION OF HYDROGENIC BOUND-FREE RADIAL INTEGRALS USING
C           RECURRENCE RELATIONS.
C-----------------------------------------------------------------------
C UNIX-IDL PORT:
C
C AUTHOR:  WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C
C DATE:    4TH JULY 1996
C
C VERSION: 1.1                          DATE: 04-07-96
C MODIFIED: WILLIAM OSBORN
C               - FIRST VERSION.
C
C VERSION: 1.2                          DATE: 19-12-01
C MODIFIED: Martin O'Mullane
C               - Removed junk from > column 72.
C
C VERSION: 1.3                          DATE: 16-05-07
C MODIFIED: Allan Whiteford
C               - Modified comments as part of subroutine documentation
C                 procedure.
C
C-----------------------------------------------------------------------
       IMPLICIT REAL*8(A-H,O-Z)
       SC=64.0
       SCL=0.015625
       EN=N
       EN2=EN*EN
       EK2=-E2
       V=1.0+EN2*EK2
       EK=DSQRT(EK2)
       V=V*V
       U=8.0*EN2/V
       P=1.0
       JS=0
       SC2=SC*SC
       SCL2=SCL*SCL
       DO 5 I=1,N
       EI=I
       P=P*U*(1.0+EI*EI*EK2)/(EI*(2.0*EI-1.0))
       AP=DABS(P)
       IF(SCL2.LE.AP)GO TO 5
       JS=JS-1
       P=SC2*P
    5  CONTINUE
       IF(EK.GT.0.04D0)GO TO 6
       P=EN*P
       GO TO 7
    6  P=EN*P/(1.0-DEXP(-6.283185/EK))
    7  IF(EK.GT.1.0D-5)GO TO 8
       U=-2.0*EN
       GO TO 9
    8  U=-2.0*DATAN(EN*EK)/EK
    9  T2=7.089815*DSQRT(P)*DEXP(U)/V
       V=1.0+EN2*EK2
       IF(L2.EQ.L+1)GO TO 11
       IF(L2.EQ.L-1)GO TO 20
```

```
    RD2FS=0.0
    GO TO 50
11  U=(2.0*EN-1.0)*V
    U=DSQRT(U)
    T3=0.5*U*T2
    NU=N-2
    IF(L-NU)14,13,12
12  T3=T2
13  GO TO 40
14  DO 16 I=L2,NU
    LI=NU-I+L
    EL1=LI+1
    EL2=LI+2
    ES=EL2*EL2
    T1=T2
    T2=T3
    T3=(4.0*(EN2-ES)+EL2*(2.0*EL2-1.0)*V)*T2-2.0*EN*U*T1
    U=(EN2-EL1*EL1)*(1.0+ES*EK2)
    U=DSQRT(U)
    T3=T3/(2.0*EN*U)
    AT3=DABS(T3)
    IF(AT3.LE.SC)GO TO 16
    JS=JS+1
    T3=SCL*T3
    T2=SCL*T2
16  CONTINUE
    GO TO 40
20  EN1=N-1
    U=V/(1.0+EN1*EN1*EK2)
    T2=DSQRT(U)*T2/(2.0*EN)
    U=(2.0*EN-1.0)*(1.0+(EN-2.0)*(EN-2.0)*EK2)
    U=DSQRT(U)
    T3=(4.0+EN1*V)*(2.0*EN-1.0)*T2/(2.0*EN*U)
    NU=N-3
    IF(L-NU-1)24,23,22
22  T3=T2
23  GO TO 40
24  DO 26 I=L,NU
    LI=NU-I+L
    EL=LI
    EL1=LI+1
    ES=EL1*EL1
    T1=T2
    T2=T3
    T3=(4.0*(EN2-ES)+EL1*(2.0*EL1+1.0)*V)*T2-2.0*EN*U*T1
    U=(EN2-ES)*(1.0+EL*EL*EK2)
    U=DSQRT(U)
    T3=T3/(2.0*EN*U)
    AT3=DABS(T3)
    IF(AT3.LE.SC)GO TO 26
    JS=JS+1
    T3=SCL*T3
    T2=SCL*T2
26  CONTINUE
40  RJS=JS
    RD2FS=EN2*EN2*T3*T3*4096.0**RJS
50  RETURN
   END
   INTEGER             L,           L2,          N
   REAL*8              E2
```

3.138 voigt: Subroutine voigt from library adas2xx

```
       subroutine voigt ( iprofile, y0, yl, yt, x, y, iz, w2f, wid,
     &     fact, xmin, xmax)
C
C-----------------------------------------------------------------
C
C *****************FORTRAN77 SUBROUTINE: VOIGT *************
C
C  ORIGINAL NAME: VOIGT.BAS
C (DEVELOPED BY KURT BEHRINGER)
C
C  VERSION: 1.0
C
C  PURPOSE: COMPUTES LINE PROFILES
C
C  CALLING PROGRAM: ESCAPE.FOR
C
C  INPUT:
C
C
C
C  ROUTINES:
C ROUTINE SOURCE BRIEF DESCRIPTION
C -------------------------------------------------
C
C
C  AUTHOR: K.H.BEHRINGER (IPF, UNIVERSITY OF STUTTGART)
C
C  DATE: 31/4/98
C
C  UPDATE:
C
C  VERSION: 1.1 DATE: 18-06-98
C  MODIFIED: STUART LOCH
C  - CONVERTED TO FORTRAN FOR ADAS.
C
C  VERSION: 1.2 DATE: 26-11-98
C  MODIFIED: STUART LOCH
C  - CONVOLVED PROFILES ARE NORMALISED TO
C    ACCOUNT FOR NUMERICAL INACCURACIES WHICH
C    GAVE ESCAPE FACTORS GREATER THAN ONE
C  VERSION: 1.3                                          DATE: 19-02-99
C  MODIFIED: STUART lOCH
C               -DIMENSIONS OF X,Y, YT, YL AND MODPROF INCREASED FROM 1000
C                TO 3000, TO ACCOMODATE POSSIBLE INCREASES IN INTEGRATION
C                LIMIT OF LINE PROFILE

c-----------------------------------------------------------------
      INTEGER             IPROFILE,    IZ
      REAL*8              FACT,        W2F,         WID,         X(3000)
      REAL*8              XMAX,        XMIN,        Y(3000),     Y0
      REAL*8              YL(3000),    YT(3000)
```

4 Subroutine library adas3xx

4.1 abi: Subroutine abi from library adas3xx

```
       FUNCTION ABI(OA,YA,FA,A,B,N)
C
       IMPLICIT REAL*8(A-H,O-Z)
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 FUNCTION: ABI *************************
C
C-----------------------------------------------------------------------
C  PURPOSE: EVALUATE INNER INTEGRALS FOR RATE COEFFICIENTS.
C
C  ABI IS CALLED BY QHIOCH. THE PROCEDURES ALLOW FOR PROJECTILE
C  AND TARGET MASSES.
C
C  ********** H.P. SUMMERS,JET            18 FEB 1987  ***************
C  INPUT
C      OA=VECTOR OF CROSS-SECTIONS (CM2)
C      YA=VECTOR OF REDUCED RELATIVE SPEEDS. 1ST VALUE IS AT THRESHOLD
C      FA=VECTOR OF ALPHAS  (12TH VALUE IS FOR EXTRAPOLATION, PROVIDED
C         EXPLICITLY. EXTRAPOLATION BELOW YA(1) IS BASED ON FA(1))
C         N.B. OA(1) MUST BE NON-ZERO.
C             (OA,YA AND FA ARE OF FIXED LENGTH =12)
C
C      A=LOWER INTEGRAL LIMIT
C      B=UPPER INTEGRAL LIMIT
C      N=NUMBER OF CROSS-SECTIONS
C  OUTPUT
C      ABI=DEFINITE INTEGRAL.
C
C NOTES: THIS ROUTINE IS NOT YET PROPERLY ANNOTATED
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 18-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - PUT UNDER SCCS CONTROL
C
C VERSION: 1.2 DATE: 16-05-07
C MODIFIED: Allan Whiteford
C     - Updated comments as part of subroutine documentation
C             procedure.
C
C-----------------------------------------------------------------------
C
      INTEGER             N
      REAL*8              A,           B,           FA(24),      OA(24)
      REAL*8              YA(24)
```

4.2 abinew: Subroutine abinew from library adas3xx

```
       FUNCTION ABINEW(OA,YA,FA,A,B,N)
C
       IMPLICIT REAL*8(A-H,O-Z)
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 FUNCTION: ABINEW ***********************
C
C-----------------------------------------------------------------------
C  PURPOSE: EVALUATE INNER INTEGRALS FOR RATE COEFFICIENTS.
C
C  ABINEW IS CALLED BY QH & QHE.  THE PROCEDURES ALLOW FOR PROJECTILE
C  AND TARGET  MASSES.
C
C  INPUT
C      OA=VECTOR OF CROSS-SECTIONS (CM2)
C      YA=VECTOR OF REDUCED RELATIVE SPEEDS. 1ST VALUE IS AT THRESHOLD
C      FA=VECTOR OF ALPHAS  (12TH VALUE IS FOR EXTRAPOLATION, PROVIDED
C         EXPLICITLY. EXTRAPOLATION BELOW YA(1) IS BASED ON FA(1))
C         N.B. OA(1) MUST BE NON-ZERO.
C             (OA,YA AND FA ARE OF FIXED LENGTH =24)
C
C      A=LOWER INTEGRAL LIMIT
C      B=UPPER INTEGRAL LIMIT
C      N=NUMBER OF CROSS-SECTIONS
C  OUTPUT
C      ABINEW=DEFINITE INTEGRAL.
C
C  ********** H.P. SUMMERS,JET            18 FEB 1987  ****************
C  **********                         COR. 31 JUL 1990  ****************
C  **********                              15 JUL 1991  REORDER EVAL AT
C                                                       LABEL 50
C
C NOTES: THIS ROUTINE IS NOT YET PROPERLY ANNOTATED
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 16-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST VERSION
C
C VERSION: 1.2 DATE: 16-05-07
C MODIFIED: Allan Whiteford
C     - Updated comments as part of subroutine documentation
C             procedure.
C
C-----------------------------------------------------------------------
      INTEGER             N
      REAL*8              A,           B,           FA(24),      OA(24)
      REAL*8              YA(24)
```

4.3 bnqctb: Subroutine bnqctb from library adas3xx

```
       SUBROUTINE BNQCTB(Z0,Z1,NMIN,NMAX,IMAX,NREP,NBEAM,BMENA,BMFRA,
     &                   CXMEMB,IBLOCK,QTHREP,ALPHA)
C
       IMPLICIT REAL*8(A-H,O-Z)
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: BNQCTB ********************
C
C -----------------------------------------------------------------------
C  PURPOSE: CALCULATE THEORETICAL CHARGE EXCHANGE RATE COEFFICIENTS
C  FROM NEUTRAL HYDROGEN.
C
C  RATE DATA IS RETURNED TO REPRESENTATIVE N-SHELLS FOR USE BY BUNDLE-N
C  CODES.
C
C  INPUT FROM ARCHIVED DATASET IS ON UNIT 11.
C
C  THE NAME OF THE SELECTED DATASET IS CONTAINED IN: 'CXMEMB'
C
C  AND IS OPENED IN THE SUBROUTINE.
C
C  THIS VERSION USES   '1989 RESTRUCTURED DATA'   MEMBERS WITH THE
C  CHANGED L-FITTING PARAMETERS
C  THE NEW PARAMETERS ARE TRANSFERED IN   COMMON /LFIT89/
C
C  THE SUBROUTINE IS A DEVELOPMENT OF QCHEX, NEWCX2, NCHEX2 ETC.
C  ORIGINALLY WRITTEN BY J. SPENCE.  THIS VERSION ECONOMISES ON
C  SUBROUTINES.
C
C  INPUT
C      Z0=TARGET ION NUCLEAR CHARGE
C      Z1=RECOMBINING TARGET ION CHARGE
C      NMIN=LOWEST REPRESENTATIVE N-LEVEL OF TARGET
C      NMAX=HIGHEST REPRESENTATIVE N-LEVEL OF TARGET
C      IMAX=NUMBER OF REPRESENTATIVE LEVELS
C      NREP(I)=REPRESENTATIVE N-LEVELS
C      NBEAM=NUMBER OF ENERGY COMPONENTS IN NEUTRAL HYDROGEN BEAM
C      BMENA(J)=BEAM ENERGY COMPONENTS (EV/AMU)
C      BMFRA(J)=BEAM FRACTIONS IN ENERGY COMPONENTS
C      CXMEMB=DATA SET NAME OF CHARGE EXCHANGE DATA SET.
C      IBLOCK=1 SELECT UDW METHOD OR 1ST DATA BLOCK
C             =2 SELECT CCAO METHOD OR 2ND DATA BLOCK
C             =3 SELECT CTMC METHOD OR 3RD DATA BLOCK
C             =4 SELECT CCMO METHOD OR 4TH DATA BLOCK
C
C  OUTPUT
C      QTHREP(I)=MEAN RATE COEFFICIENTS FOR REPRESENTATIVE
C                N-LEVELS  (AVERAGED OVER BEAM FRACTIONS) (CM3 SEC-1)
C      ALPHA=SIZE OF 1/N**ALPHA TAIL FOR CH.EXCH X-SECT.
C
C  ********** H.P.SUMMERS, JET            13 DEC 1989  ***************
C -----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
C
C UPDATE:  19/01/94 - JONATHAN NASH - TESSELLA SUPPORT SERVICES PLC
C
C          THE FOLLOWING MODIFICATIONS HAVE BEEN MADE TO THE SUBROUTINE:
```

```
C
C          1) THE COMPLETE CHARGE EXCHANGE DATA SET NAME IS NOW PASSED
C             INTO THE ROUTINE RATHER THAN JUST THE MEMBER NAME.
C
C          2) THE ROUTINE HAS BEEN UPGRADED TO READ NEW ADF01 FORMAT.
C
C  NOTES:  NO ATTEMPT HAS BEEN MADE TO RESTRUCTURE THE ROUTINE. RATHER
C          THE MINIMUM AMOUNT OF WORK TO INTEGRATE THE ROUTINE INTO
C          ADAS310 HAS BEEN COMPLETED.
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 16-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST VERSION
C
C VERSION: 1.2                          DATE: 17-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - ADDED "STATUS='UNKNOWN'" TO OPEN STATEMENT
C
C VERSION: 1.3                          DATE: 22-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - REPLACED CALLS TO NAG ROUTINE E02BBF WITH ADAS ROUTINE
C                 DXNBBF
C
C VERSION: 1.4                          DATE: 23-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - REPLACED CALLS TO NAG ROUTINE E01BAF WITH ADAS ROUTINE
C                 DXNBAF
C
C VERSION: 1.5                          DATE: 23-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - RELABELLED LOOP COUNTERS FOR LOOPS 176 AND 177
C
C VERSION: 1.6                          DATE: 24-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - RENAMED NBENG TO NBENG2 TO AVOID CONFUSION WITH
C                 OTHER NBENG IN OTHER ROUTINES
C                 REMOVED SUPERFLUOUS VARIABLES
C
C VERSION: 1.7                          DATE: 14-10-96
C MODIFIED: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C               - CORRECTED SECOND CALL TO DXNBAF - IT WAS USING XSA AND
C                 YSA RATHER THAN XSA AND ZSA
C
C VERSION: 1.8 DATE: 09-04-98
C MODIFIED: HARVEY ANDERSON ( UNIVERSITY OF STRATHCLYDE )
C - CHANGED VARIBLE MXE FROM 24 TO 40.
C - INCREASED SIZE OF ARRAYS ASSOCIATED WITH THE
C   ROUTINES DXNBAF AND DXNBBF.
C - REPLACED NUMERICAL VALUE WITH THE PARAMETER
C   MXE IN THE IF STATEMENT WHICH TESTS TO ENSURE
C   THAT THE NUMBER OF BEAM ENERGIES READ FROM
C   INPUT FILE IS NOT GREATER THE ARRAY DIMMENSIONS
C   OF THE RELEVANT ARRAYS.
C
C VERSION: 1.9 DATE: 23-06-98
C MODIFIED: RICHARD MARTIN
C  -CORRECTED SCCS ERROR.
C
C VERSION: 1.10 DATE: 07-07-2004
```

```
C MODIFIED: ALLAN WHITEFORD
C  -CHANGED CALLS FROM DXNB{A,B}F TO XXNB{A,B}F
C
C VERSION: 1.11 DATE: 16-05-07
C MODIFIED: Allan Whiteford
C     - Updated comments as part of subroutine documentation
C             procedure.
C
C VERSION  : 1.12
C DATE     : 22-05-2007
C MODIFIED : Martin O'Mullane
C              - Remove unused m-subshell data possibility and
C                use xxdata_01 to access adf01 data.
C
C-----------------------------------------------------------------------
C
C          (I*4)  MXE       = MAXIMUM NO. OF ENERGIES.
C          (I*4)  MXN       = MAXIMUM NO. OF N SHELLS.
C          (I*4)  IZR       = ION CHARGE OF RECEIVER.
C          (I*4)  IZD       = ION CHARGE OF DONOR.
C          (I*4)  INDD      = DONOR STATE INDEX.
C          (I*4)  NBENG2    = NUMBER OF ENERGIES READ.
C          (I*4)  NMINF     = LOWEST N-SHELL FOR WHICH DATA READ.
C          (I*4)  NMAXF     = HIGHEST N-SHELL FOR WHICH DATA READ.
C
C          (L*4)  LPARMS    = FLAGS IF L-SPLITTING PARAMETERS PRESENT.
C                             .TRUE.  => L-SPLITTING PARAMETERS PRESENT.
C                             .FALSE  => L-SPLITTING PARAMETERS ABSENT.
C          (L*4)  LSETL     = FLAGS IF L-RESOLVED DATA PRESENT.
C                             .TRUE.  => L-RESOLVED DATA PRESENT.
C                             .FALSE  => L-RESOLVED DATA ABSENT.
C          (L*4)  LSETM     = FLAGS IF M-RESOLVED DATA PRESENT.
C                             .TRUE.  => M-RESOLVED DATA PRESENT.
C                             .FALSE  => M-RESOLVED DATA ABSENT.
C
C          (C*80) TITLE     = NOT SET - TITLE FOR DATA SOURCE.
C          (C*2)  SYMBR     = RECEIVER ION ELEMENT SYMBOL.
C          (C*2)  SYMBD     = DONOR ION ELMENT SYMBOL.
C
C          (I*4)  LFORMA()  = PARAMETERS FOR CALCULATING L-RES X-SEC.
C                             DIMENSION: MXE
C
C          (R*8)  BENGY()   = COLLISION ENERGIES.
C                             UNITS: EV/AMU (READ AS KEV/AMU)
C                             DIMENSION: MXE
C          (R*8)  ALPHAA()  = EXTRAPOLATION PARAMETER ALPHA.
C                             DIMENSION: MXE
C          (R*8)  XLCUTA()  = PARAMETERS FOR CALCULATING L-RES X-SEC.
C                             DIMENSION: MXE
C          (R*8)  PL2A()    = PARAMETERS FOR CALCULATING L-RES X-SEC.
C                             DIMENSION: MXE
C          (R*8)  PL3A()    = PARAMETERS FOR CALCULATING L-RES X-SEC.
C                             DIMENSION: MXE
C          (R*8)  XTOT()    = TOTAL CHARGE EXCHANGE CROSS-SECTION.
C                             UNITS: CM2
C                             DIMENSION: MXE
C
C          (R*8)  XSIGN(,)  = N-RESOLVED CHARGE EXCHANGE CROSS-SECTIONS.
C                             UNITS: CM2
C                             1ST DIMENSION: MXE
C                             2ND DIMENSION: MXN
```

```
C          (R*8)  XSIGL(,)  = L-RESOLVED CHARGE EXCHANGE CROSS-SECTIONS.
C                              UNITS: CM2
C                              1ST DIMENSION: MXE
C                              2ND DIMENSION: (MXN*(MXN+1))/2
C          (R*8)  XSIGM(,)  = M-RESOLVED CHARGE EXCHANGE CROSS-SECTIONS.
C                              UNITS: CM2
C                              1ST DIMENSION: MXE
C                              2ND DIMENSION: (MXN*(MXN+1)*(MXN+2))/6
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      CHARACTER*80        CXMEMB
      INTEGER             IBLOCK,      IMAX,        NBEAM,       NMAX
      INTEGER             NMIN,        NREP(31)
      REAL*8              ALPHA,       BMENA(6),    BMFRA(6)
      REAL*8              QTHREP(31),  Z0,          Z1
```

4.4 bornp1: Subroutine bornp1 from library adas3xx

```
       SUBROUTINE BORNP1(NMIN,NMAX,NLOW,LLOW,NUP,LUP,LAM,ANS,QA,IOPT,
     &IMAX1)
       IMPLICIT REAL*8(A-H,O-Z)
C-----------------------------------------------------------------------
C  PURPOSE: FIRST STAGE OF BORN X-SECT CALCULATION WITH DIRECTIONAL
C  BEAMS CAUSING TRANSITIONS BETWEEN HYDROGEN STARK/ZEEMAN STATES.
C
C  RADIAL INTEGRALS OF FORM P(N1,L1)*P(N2,L2)*J(LAM,Q*R) ARE EVALUATED
C  FOR ALL HYDROGENIC ORBITALS WITH NMIN<=NMAX AND N1<N2 FOR A SET
C  OF Q'S SPANNING THE EXTENT OF THE GENERALISED OSCILLATOR STRENGTH
C
C  CALLING ROUTINE WITH IOPT=1 PREPARES LOOKUP TABLES. CALL WITH IOPT=2
C  RETURNS VALUES.
C
C  MATRIX ELEMENT EVALUATION USES POWER SERIES ROUTINES.
C
C  STACK INTEGRALS OVER A GRID OF MOMENTUM CHANGE FROM 10**-2 TO 10**2
C  EQUALLY SPACED IN THE LOGARITHM (IMAX INTERVALS, IMAX+1 VALUES)
C
C  EVALUATE MULTIPOLES UP TO LAMMX=2
C
C  USE POINTER VECTORS FOR RAPID LOOKUP.
C
C  THE MAIN CALLING ROUTINE MUST HAVE THE LINE
C      CALL GAMAF(200)
C  BEFORE CALL TO BORNP1
C
C  ******** H.P. SUMMERS, JET          17 OCT 1988   *************
C  INPUT
C      NMIN=LOWEST N-SHELL
C      NMAX=HIGHEST N-SHELL
C      NLOW=LOWER N FOR SELECTED TRANSITION
C      LLOW=LOWER L FOR SELECTED TRANSITION
C      NUP=UPPER N FOR SELECTED TRANSITION
C      LUP=UPPER L FOR SELECTED TRANSITION
C      LAM=REQUIRED MULTIPOLE (0<=LAM<=LAMMX)
C      IOPT=1  PREPARE LOOKUP STACKS (ONLY NMIN,NMAX PARAMETERS USED)
C          =2  SUPPLY ANSWERS FOR SPECIFIED NLOW,LLOW,NUP,LUP,LAM CASE
C  OUTPUT
C      ANS(I)=BORN APPROXIMATION RESULT FOR I=1,IMAX+1
C      QA(I)=TRANSFERED MOMENTUM VECTOR
C      IMAX1=IMAX+1
C
C-----------------------------------------------------------------------
C
C ADAS305 version. Developed from JETSHP.STARK.FORT (H P Summers).
C
C VERSION  : 1.1
C DATE     : 24-02-2005
C MODIFIED : Martin O'Mullane
C             - First version.
C             - Change dimensions from 200 to 500.
C
C VERSION  : 1.2
C DATE     : 16-05-2007
C MODIFIED : Allan Whiteford
C             - Updated comments as part of subroutine documentation
C               procedure.
C
```

```
C-----------------------------------------------------------------------
      INTEGER             IMAX1,       IOPT,        LAM,          LLOW
      INTEGER             LUP,         NLOW,        NMAX,         NMIN
      INTEGER             NUP
      REAL*8              ANS(41),     QA(41)
```

4.5 bornp2: Subroutine bornp2 from library adas3xx

```
       SUBROUTINE BORNP2(NMAX,LLOW,LAM,LUP,MLLOW,NU,MLUP,ANS,IOPT)
       IMPLICIT REAL*8(A-H,O-Z)
C-----------------------------------------------------------------------
C  PURPOSE: SECOND STAGE OF BORN X-SECT CALCULATION WITH DIRECTIONAL
C  BEAMS CAUSING TRANSITIONS BETWEEN HYDROGEN STARK/ZEEMAN STATES.
C
C  ANGULAR INTEGRALS OF FORM ?????????????????????????? ARE EVALUATED
C  FOR ALL ORBITAL ANGULAR MOMENTA  0<=L<=NMAX-1
C
C  CALLING ROUTINE WITH IOPT=1 PREPARES LOOKUP TABLES. CALL WITH IOPT=2
C  RETURNS VALUES.
C
C  EVALUATE MULTIPOLES UP TO LAMMX=2
C
C  USE POINTER VECTORS FOR RAPID LOOKUP.
C
C  THE MAIN CALLING ROUTINE MUST HAVE THE LINE
C      CALL GAMAF(200)
C  BEFORE CALL TO BORNP2
C
C  ******** H.P. SUMMERS, JET          17 OCT 1988   *************
C  INPUT
C      NMAX=HIGHEST N-SHELL FOR TABLE PREPARATION
C      LLOW=LOWER L FOR SELECTED TRANSITION
C      LAM=MULTIPOLE (0<=LAM<=LAMMX)
C      LUP=UPPER L FOR SELECTED TRANSITION
C      MLLOW=AZIMUTHAL QUANTUM NUMBER FOR LOWER L
C      NU=MULTIPOLE AZIMUTHAL COMPONENT
C      MLUP=AZIMUTHAL QUANTUM NUMBER FOR UPPER L
C      IOPT=1  PREPARE LOOKUP STACKS (ONLY NMAX PARAMETER USED)
C          =2  SUPPLY ANSWERS FOR SPECIFIED LLOW,LAM,LUP,MLLOW,NU,MLUP
C  OUTPUT
C      ANS(I)=RESULT
C-----------------------------------------------------------------------
C
C ADAS305 version. Developed from JETSHP.STARK.FORT (H P Summers).
C
C VERSION  : 1.1
C DATE     : 24-02-2005
C MODIFIED : Martin O'Mullane
C             - First version.
C             - Change dimensions from 200 to 500.
C
C VERSION  : 1.2
C DATE     : 16-05-2007
C MODIFIED : Allan Whiteford
C             - Updated comments as part of subroutine documentation
C               procedure.
C
C-----------------------------------------------------------------------
      INTEGER             IOPT,        LAM,         LLOW,        LUP
      INTEGER             MLLOW,       MLUP,        NMAX,        NU
      REAL*8              ANS
```

4.6 c1bsig: Subroutine c1bsig from library adas3xx

```
      SUBROUTINE C1BSIG( NENRGY , NSHELL , NENER ,
     &                   INSEL  , ILSEL  ,
     &                   IEDATA , NDATA  ,
     &                   XTOT   , XSIGN  , XSIGL ,
     &                   ALPHAA ,
     &                   SIGA
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: C1BSIG *********************
C
C  PURPOSE:  TO EXTRACT VALID CROSS-SECTIONS FROM INPUT CHARGE-EXCHANGE
C            FILE FOR A GIVEN SUB-BLOCK.
C
C  CALLING PROGRAM: ADAS301
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  NENRGY   = NUMBER OF TABULATED INPUT ENERGIES.
C  INPUT : (I*4)  NSHELL   = NO. OF TABULATED PRINCIPAL QUANTUM N-SHELLS
C
C  INPUT : (I*4)  NENER    = NUMBER OF VALID ENERGIES/VEL. FOR SUB-BLOCK
C  INPUT : (I*4)  INSEL    = SELECTED INPUT PRINCIPAL QUANTUM N-SHELL
C                            ( 0 => TOTAL CROSS SECTION).
C  INPUT : (I*4)  ILSEL    = SELECTED INPUT DATA L QUANTUM SHELL
C                            ( -1   => TOTAL CROSS-SECTION FOR N SHELL)
C  INPUT : (I*4)  IMSEL    = SELECTED INPUT DATA M QUANTUM SHELL
C                            ( -1   => TOTAL CROSS-SECTION FOR NL SHELL)
C
C  INPUT : (I*4)  IEDATA() = INDEX RANGE FOR VALID QUANTUM NUMBERS IN
C                            'XSIGN(,)'.
C                             DIMENSION: 1 => LOWER INDEX BOUND
C                                        2 => UPPER INDEX BOUND
C  INPUT : (I*4)  NDATA()  = INDEX RANGE FOR VALID INPUT ENERGIES/VELS.
C                            IN 'XTOT()', 'XSIGN(,)','ALPHAA()'
C                             DIMENSION: 1 => LOWER INDEX BOUND
C                                        2 => UPPER INDEX BOUND
C
C  INPUT : (R*8)  XTOT()   = TOTAL CROSS SECTIONS (UNITS: cm**2)
C                            1st DIMENSION: ENERGY INDEX
C  INPUT : (R*8)  XSIGN(,) = CROSS SECTIONS FOR EACH N-SHELL
C                                   (UNITS: cm**2)
C                            1st DIMENSION: ENERGY INDEX
C                            2nd DIMENSION: QUANTUM N-SHELL INDEX
C  INPUT : (R*8)  XSIGL(,,)=INPUT DATA FILE: L-RESOLVED CROSS-SECTIONS.
C                           1st DIMENSION: ENERGY INDEX
C                           2ND DIMENSION: INDEXED BY FUNCTION I4IDFL.
C                           3rd DIMENSION: DATA SUB-BLOCK INDEX
C  INPUT : (R*8)  XSIGM(,,)=INPUT DATA FILE: M-RESOLVED CROSS-SECTIONS.
C                           1st DIMENSION: ENERGY INDEX
C                           2ND DIMENSION: INDEXED BY FUNCTION I4IDFM.
C                           3rd DIMENSION: DATA SUB-BLOCK INDEX
C  INPUT : (R*8)  ALPHAA() =
C                            1st DIMENSION: ENERGY INDEX
C
C  OUTPUT: (R*8)  SIGA()   = VALID CROSS SECTIONS READ FROM INPUT FILE
C                            FOR PRINCIPAL QUATUM NUMBER 'INSEL' AND
C                            GIVEN SUB-BLOCK. (UNITS: cm**2)
C                            1st DIMENSION: ENERGY INDEX
```

```
C
C          (I*4)  IE       = ARRAY INDEX: ENERGY INDEX
C          (I*4)  I1ST     = ARRAY INDEX: FIRST VALID ENERGY INDEX - 1
C          (I*4)  IN       = 'NDATA(2)' - IF 'INSEL > NDATA(2)'
C
C          (R*8)  AVAL     = 'NDATA(2)/INSEL' -  IF 'INSEL > NDATA(2)'
C          (R*8)  ZERO     = PARAMETER = EFFECTIVE ZERO (1.0D-72)
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          -------------------------------------------------------
C          I4IDFL     ADAS      PROVIDES UNIQUE INDEX GIVEN N AND L
C          I4IDFM     ADAS      PROVIDES UNIQUE INDEX GIVEN N, L AND M
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 4569
C
C DATE:    07/02/91
C
C UPDATE:  19/04/95  H P SUMMERS - EXTENSION TO INCLUDE L AND M
C                                  SUB-SHELL CROSS-SECTIONS
C
C VERSION  : 1.2
C DATE     : 22-05-2007
C MODIFIED : Martin O'Mullane
C              - Remove unused m-subshell data possibility.
C
C-----------------------------------------------------------------------
      INTEGER             IEDATA(2),   ILSEL,       INSEL
      INTEGER             NDATA(2),    NENER,       NENRGY,      NSHELL
      REAL*8              ALPHAA(NENRGY),           SIGA(NENRGY)
      REAL*8              XSIGL(NENRGY,(NSHELL*(NSHELL+1))/2)
      REAL*8              XSIGN(NENRGY,NSHELL),     XTOT(NENRGY)
```

4.7 c1spln: Subroutine c1spln from library adas3xx

```
      SUBROUTINE C1SPLN(         LOSEL ,
     &                   NENER , IEVAL , NPSPL  ,
     &                   ENERA , EOA   , EOSA   ,
     &                   SIGA  , SIGOA , SIGOSA ,
     &                   LERNG
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: C1SPLN *********************
C
C  (IDENTICAL TO: B1SPLN (EXCEPT SOME VARIABLE NAMES ARE CHANGED))
C
C  PURPOSE:
C         1) PERFORMS CUBIC SPLINE ON LOG(ENERGY) VERSUS LOG(X-SECTION)
C            INPUT DATA, ('ENERA' VERSUS 'SIGA' , NENER DATA PAIRS),
C            FOR A GIVEN SUB-BLOCK.
C
C         2) INTERPOLATES 'IEVAL' X-SECT. VALUES USING  ABOVE  SPLINES
C            AT ENERGIES READ IN FROM ISPF PANELS FOR TABULAR  OUTPUT.
C            (ANY ENERGIES VALUES WHICH REQUIRED EXTRAPOLATION ARE SET
C             TO ZERO).
C                 - THIS STEP ONLY TAKES PLACE IF 'LOSEL=.TRUE.' -
C
C         3) INTERPOLATES 'NPSPL' X-SECT VALUES USING ABOVE SPLINES AT
C            ENERGIES EQUI-DISTANCE  ON RANGE OF LOG(ENERGIES)  STORED
C            IN INPUT 'ENERA' ARRAY.
C
C  CALLING PROGRAM: ADAS301
C
C
C  SUBROUTINE:
C
C  INPUT : (L*4)  LOSEL   = .TRUE.  => CALCULATE X-SECS FOR INPUT ENGYS.
C                                      READ FROM ISPF PANEL.
C                           .FALSE. => - DO NOT DO THE ABOVE -
C
C  INPUT : (I*4)  NENER   = INPUT DATA FILE: NO. OF VALID ENERGY/X-SECT.
C                           PAIRS READ FOR THE SUB-BLOCK BEING ASSESSED
C  INPUT : (I*4)  IEVAL   = NUMBER OF  ISPF  ENTERED  ENERGY  VALUES  AT
C                           WHICH INTERPOLATED X-SEC VALUES ARE REQUIRED
C                           FOR TABULAR OUTPUT.
C  INPUT : (I*4)  NPSPL   = NUMBER OF  SPLINE  INTERPOLATED  ENGY/X-SECT
C                           REQUIRED FOR GRAPHICAL DISPLAY.
C
C  INPUT : (I*4)  ENERA() = INPUT DATA FILE: ENERGIES (EV/AMU)
C  INPUT : (I*4)  EOA()   = ISPF PANEL ENTERED ENERGIES (EV/AMU)
C  OUTPUT: (I*4)  EOSA()  = 'NPSPL' ENERGIES FOR GRAPHICAL OUTPUT
C                           (EV/AMU).
C
C  INPUT : (R*8)  SIGA()  = INPUT DATA FILE: SELECTED SUB-BLOCK -
C                           X-SECTION VALUES AT 'ENERA()'. (CM**2)
C  OUTPUT: (I*4)  SIGOA() = SPLINE INTERPOLATED X-SEC VALUES AT 'EOA()'
C                           (EXTRAPOLATED VALUES = 0.0).
C  OUTPUT: (R*8)  SIGOSA()= SPLINE INTERPOLATED X-SEC VALUES AT 'EOSA()'
C
C  OUTPUT: (L*4)  LERNG() = .TRUE.  => OUTPUT   SPLINE    VALUE    WAS
C                                      INTERPOLATED FOR 'DLOG(EOA()'.
C                           .FALSE. => OUTPUT   SPLINE    VALUE    WAS
C                                      EXTRAPOLATED FOR 'DLOG(EOA()'.
```

```
C                                     (NOTE: 'YOUT()=0' AS 'IOPT < 0').
C
C          (I*4)  NIN     = PARAMETER = MAX. NO. OF  INPUT  ENGY/X-SEC
C                                       PAIRS MUST BE >= 'NENER'
C          (I*4)  NOUT    = PARAMETER = MAX. NO. OF 'OUTPUT ENGY/X-SEC
C                                       PAIRS MUST BE >= 'IEVAL' &'NPSPL'
C          (R*8)  ZERO    = PARAMETER = EFFECTIVE ZERO (1.0D-72)
C          (R*8)  ZEROL   = PARAMETER = LN(ZERO) APPROX.  = -165.7
C
C          (I*4)  IARR    = ARRAY SUBSCRIPT USED FOR ENGY/X-SEC PAIRS
C          (I*4)  IOPT    = DEFINES THE BOUNDARY DERIVATIVES FOR THE
C                           SPLINE ROUTINE 'XXSPLE', SEE 'XXSPLE'.
C                           (VALID VALUES = <0, 0, 1, 2, 3, 4)
C
C          (R*8)  ESTEP   = THE SIZE OF STEP BETWEEN 'XOUT()' VALUES FOR
C                           GRAPHICAL  OUTPUT  ENGY/X-SEC  PAIRS  TO  BE
C                           CALCULATED USING SPLINES.
C
C          (L*4)  LSETX   = .TRUE.  => SET UP SPLINE PARAMETERS RELATING
C                                      TO 'XIN' AXIS.
C                           .FALSE. => DO NOT SET UP SPLINE PARAMETERS
C                                      RELATING TO 'XIN' AXIS.
C                                      (I.E. THEY WERE SET IN A PREVIOUS
C                                            CALL )
C                           (VALUE SET TO .FALSE. BY 'XXSPLE')
C
C          (R*8)  XIN()   = LOG( 'ENERA()' )
C          (R*8)  YIN()   = LOG( 'SIGA()' )
C          (R*8)  XOUT()  = LOG(ENERGIES AT WHICH SPLINES REQUIRED)
C          (R*8)  YOUT()  = LOG(OUTPUT SPLINE INTERPOLATED X-SEC VALUES)
C          (R*8)  DF()    = SPLINE INTERPOLATED DERIVATIVES
C
C          (L*4)  LDUMP() = .TRUE.  => OUTPUT SPLINE VALUE INTRPOLATED
C                                      FOR 'YOUT()'.
C                           .FALSE. => OUTPUT SPLINE VALUE EXTRAPOLATED
C                                      FOR 'YOUT()'.
C                                      (NOTE: USED AS A DUMMY ARGUMENT.
C                                             ALL VALUES WILL BE TRUE.)
C
C NOTE:
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          XXSPLE     ADAS      SPLINE SUBROUTINE (EXTENDED DIAGNOSTICS)
C          R8FUN1     ADAS      REAL*8 FUNCTION: ( X -> X )
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 4569
C
C DATE:    07/02/91
C
C UPDATE:  30/11/93 - J NASH    - ADAS91:
C          INCREASED MAX NUMBER OF INPUT ENERGIES ('NIN') FROM 24 TO 30.
C
C UPDATE:  19/04/95   H P SUMMERS - ADDED TRAP FOR ZERO INPUTS
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
```

```
INTEGER             IEVAL,       NENER,       NPSPL
LOGICAL             LERNG(IEVAL),             LOSEL
REAL*8              ENERA(NENER),             EOA(IEVAL)
REAL*8              EOSA(NPSPL), SIGA(NENER), SIGOA(IEVAL)
REAL*8              SIGOSA(NPSPL)
```

4.8 c2chkb: Subroutine c2chkb from library adas3xx

```
C
      SUBROUTINE C2CHKB( IUNIT  , NBSEL  , IBSEL  ,
     &                   IZ0IN  , IZ0    ,
     &                   LOPEN  , IRCODE
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: C2CHKB *********************
C
C  PURPOSE: TO CHECK THE SELECTED BLOCK (IBSEL) OF DATA  EXISTS  IN  THE
C           INPUT DATA SET AND IF SO IT REPRESENTS THE ENTERED VALUES OF
C           'IZ0IN' (NUCLEAR CHARGE OF SELECTED IONISING ION ELEMENT).
C
C           IT ALSO CLOSES THE INPUT DATA SET ALLOCATION IF OPEN.
C
C  CALLING PROGRAM: SSIA
C
C  SUBROUTINE:
C
C  INPUT : (I*4)   IUNIT   = UNIT TO WHICH INPUT DATA SET IS ALLOCATED
C  INPUT : (I*4)   NBSEL   = TOTAL NUMBER OF DATA-BLOCKS READ FROM INPUT
C                            DATA SET.
C  INPUT : (I*4)   IBSEL   = INDEX OF DATA-BLOCK SELECTED FOR ANALYSIS
C
C  INPUT : (I*4)   IZ0IN   = REQUESTED: NUCLEAR CHARGE OF ELEMENT
C  INPUT : (I*4)   IZ0     = INPUT FILE: NUCLEAR CHARGE OF ELEMENT
C
C  I/O   : (L*4)   LOPEN   = INPUT : .TRUE.  => INPUT DATA SET OPEN.
C                                    .FALSE. => INPUT DATA SET CLOSED.
C                            OUTPUT: ALWAYS RETURNED AS .FALSE.
C  OUTPUT: (I*4)   IRCODE  = RETURN CODE FROM SUBROUTINE:
C                            0 => NO ERROR DETECTED.
C                            2 => DISCREPANCY BETWEEN REQUESTED ELEMENT
C                                 AND THAT IN INPUT DATA FILE.
C                            3 => SELECTED DATA-BLOCK  OUT OF RANGE  OR
C                                 DOES NOT EXIST.
C
C           (I*4)   I4UNIT  = FUNCTION (SEE ROUTINE SECTION BELOW)
C
C           (C*44)  DSNAME  = FULL MVS NAME OF DATA SET OPENED
C
C ROUTINES:
C           ROUTINE    SOURCE    BRIEF DESCRIPTION
C           ------------------------------------------------------------
C           E2FILE     ADAS      OPEN DATA SET FOR SELECTED ELEMENT
C           I4UNIT     ADAS      FETCH UNIT NUMBER FOR OUTPUT OF MESSAGES
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/37
C          JET EXT. 2520
C
C DATE:    06/06/91
C
C UPDATE:  23/04/93 - PE BRIDEN - ADAS91: ADDED I4UNIT FUNCTION TO WRITE
C                                         STATEMENTS FOR SCREEN MESSAGES
C
C UPDATE:  24/05/93 - PE BRIDEN - ADAS91: CHANGED I4UNIT(0)-> I4UNIT(-1)
C
C UNIX-IDL PORT: H.P.SUMMERS
```

```
C
C VERSION: 1.1 DATE: 19-11-96
C MODIFIED: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C     - PUT UNDER S.C.C.S. CONTROL
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IBSEL,       IRCODE,      IUNIT,       IZ0
      INTEGER             IZ0IN,       NBSEL
      LOGICAL             LOPEN
```

4.9 c2data: Subroutine c2data from library adas3xx

```
C
       SUBROUTINE C2DATA( IUNIT  , DSNAME ,
     &                    NSTORE , NEDIM  ,
     &                    NBSEL  , ISELA  ,
     &                    CPRIMY , CSECDY , CTYPE,
     &                    AMPA   , AMSA   , ALPHA  , ETHRA  ,
     &                    IEA    ,
     &                    TEEA   , SIA
     &                  )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: C2DATA *********************
C
C  PURPOSE:  TO  FETCH  DATA  FROM  INPUT ION/ATOM CROSS-SECTION
C            FILES OF TYPE ADF02.
C
C  CALLING PROGRAM: ADAS302/SSIA
C
C  DATA:
C
C           UP TO 'NSTORE' SETS (DATA-BLOCKS) OF DATA MAY BE  READ FROM
C           THE FILE - EACH BLOCK FORMING A COMPLETE SET OF  CROSS-
C           SECTION VALUES FOR GIVEN COLLISION ENERGIES.
C           EACH DATA-BLOCK  IS  ANALYSED INDEPENDENTLY OF ANY  OTHER
C           DATA-BLOCK.
C
C           THE UNITS USED IN THE DATA FILE ARE TAKEN AS FOLLOWS:
C
C           COLLISION ENERGIES  : EV/AMU
C           CROSS-SECTION       : CM**2
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  IUNIT    = UNIT TO WHICH INPUT FILE IS ALLOCATED.
C  INPUT : (C*80) DSNAME   = MVS DATA SET NAME OF DATA SET BEING READ
C
C  INPUT : (I*4)  NSTORE   = MAXIMUM NUMBER  OF  INPUT DATA-BLOCKS  THAT
C                            CAN BE STORED.
C  INPUT : (I*4)  NEDIM    = MAX NUMBER OF COLLISION ENERGIES ALLOWED
C
C  OUTPUT: (I*4)  NBSEL    = NUMBER OF DATA-BLOCKS ACCEPTED & READ IN.
C  OUTPUT: (I*4)  ISELA()  = READ - DATA-SET DATA-BLOCK ENTRY INDICES
C                            DIMENSION: DATA-BLOCK INDEX
C
C  OUTPUT: (C*5)  CPRIMY() = READ - PRIMARY SPECIES IDENTIFICATION
C                            DIMENSION: DATA-BLOCK INDEX
C  OUTPUT: (C*5)  CSECDY() = READ - SECONDARY SPECIES IDENTIFICATION
C                            DIMENSION: DATA-BLOCK INDEX
C  OUTPUT: (C*3)  CTYPE()  = READ - CROSS-SECTION TYPE
C                            DIMENSION: DATA-BLOCK INDEX
C
C  OUTPUT: (R*8)  AMPA()   = READ - PRIMARY SPECIES ATOMIC MASS NUMBER
C                            DIMENSION: DATA-BLOCK INDEX
C  OUTPUT: (R*8)  AMSA()   = READ - SECONDARY SPECIES ATOMIC MASS NUMBER
C                            DIMENSION: DATA-BLOCK INDEX
C  OUTPUT: (R*8)  ALPHA()  = READ - HIGH ENERGY EXTRAPOLATION PARM.
C                            DIMENSION: DATA-BLOCK INDEX
C  OUTPUT: (R*8)  ETHRA()  = READ - ENERGY THRESHOLD (EV)
C                            DIMENSION: DATA-BLOCK INDEX
```

```
C
C  OUTPUT: (I*4)  IEA()    = READ - NUMBER OF COLLISION ENERGIES
C                            DIMENSION: DATA-BLOCK INDEX
C
C  OUTPUT: (R*8)  TEEA(,)  = READ - COLLISION ENERGIES (UNITS: eV/AMU)
C                            1st DIMENSION: COLLISION ENERGY INDEX
C                            2nd DIMENSION: DATA-BLOCK INDEX
C
C  OUTPUT: (R*8)  SIA(,)   =READ - FULL SET OF COLLISION CROSS-
C                                    SECTION VALUES (cm**2)
C                            1st DIMENSION: COLLISION ENERGY INDEX
C                            2nd DIMENSION: DATA-BLOCK INDEX
C
C          (I*4)  I4EIZ0   = FUNCTION - (SEE ROUTINES SECTION BELOW)
C          (I*4)  I4FCTN   = FUNCTION - (SEE ROUTINES SECTION BELOW)
C          (I*4)  I4UNIT   = FUNCTION - (SEE ROUTINE SECTION BELOW)
C          (I*4)  IBLK     = ARRAY INDEX: DATA-BLOCK INDEX
C          (I*4)  ITT      = ARRAY INDEX: COLLISION ENERGY INDEX
C          (I*4)  NENUM    = NUMBER OF COLLISION ENERGIES FOR CURRENT
C                            DATA-BLOCK
C          (I*4)  IABT     = RETURN CODE FROM 'I4FCTN'
C          (I*4)  IPOS2    = GENERAL USE STRING INDEX VARIABLE
C
C          (R*8)  R8FCTN   = FUNCTION - (SEE ROUTINES SECTION BELOW)
C
C          (L*4)  LBEND    = IDENTIFIES WHETHER THE LAST OF THE  INPUT
C                            DATA SUB-BLOCKS HAS BEEN LOCATED.
C                            (.TRUE. => END OF SUB-BLOCKS REACHED)
C
C          (C*1)  CSLASH   = '/' - DELIMITER FOR 'XXHKEY'
C          (C*2)  C2       = GENERAL USE TWO BYTE CHARACTER STRING
C          (C*1)  CKEY1    = 'P'      - INPUT BLOCK HEADER KEY
C          (C*1)  CKEY2    = 'S'      - INPUT BLOCK HEADER KEY
C          (C*1)  CKEY3    = 'A'      - INPUT BLOCK HEADER KEY
C          (C*1)  CKEY4    = 'E'      - INPUT BLOCK HEADER KEY
C          (C*4)  CKEY5    = 'T'      - INPUT BLOCK HEADER KEY
C          (C*4)  CKEY6    = 'ISEL'   - INPUT BLOCK HEADER KEY
C          (C*3)  C3       = GENERAL USE THREE BYTE CHARACTER STRING
C          (C*9)  C10      = GENERAL USE NINE BYTE CHARACTER STRING
C          (C*80) C80      = GENERAL USE 80 BYTE  CHARACTER  STRING  FOR
C                            THE INPUT OF DATA-SET RECORDS.
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          XXHKEY     ADAS      OBTAIN KEY/RESPONSE STRINGS FROM TEXT
C          I4FCTN     ADAS      INTEGER*4 FUNCTION    -
C                               CONVERT CHARACTER STRING TO INTEGER
C          I4UNIT     ADAS      INTEGER*4 FUNCTION    -
C                               FETCH UNIT NUMBER FOR OUTPUT OF MESSAGES
C          R8FCTN     ADAS      REAL*8 FUNCTION       -
C                               CONVERT CHARACTER STRING TO REAL*8
C
C AUTHOR:  H. P. SUMMERS, UNIVERSITY OF STRATHCLYDE
C          JA8.08
C          TEL. 0141-553-4196
C
C DATE:    12/11/96
C
C UNIX-IDL PORT: H.P.SUMMERS
C
```

```
C VERSION: 1.1 DATE: 19-11-96
C MODIFIED: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C     - PUT UNDER S.C.C.S. CONTROL
C
C VERSION: 1.2 DATE: 14-02-97
C MODIFIED: RICHARD MARTIN
C - CHANGED INITIALISATION 'CKEY2 /'S  '/' TO 'CKEY2 /'S'/ '
C
C
C-----------------------------------------------------------------------
      CHARACTER*5         CPRIMY(NSTORE),           CSECDY(NSTORE)
      CHARACTER*3         CTYPE(NSTORE)
      CHARACTER*80        DSNAME
      INTEGER             IEA(NSTORE), ISELA(NSTORE),            IUNIT
      INTEGER             NBSEL,       NEDIM,       NSTORE
      REAL*8              ALPHA(NSTORE),            AMPA(NSTORE)
      REAL*8              AMSA(NSTORE),             ETHRA(NSTORE)
      REAL*8              SIA(NEDIM,NSTORE),        TEEA(NEDIM,NSTORE)
```

4.10 c2econ: Subroutine c2econ from library adas3xx

```
C
      SUBROUTINE C2ECON( INTYP, OUTTYP, AMD,AMR,IEVAL, EIN, EOUT )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: C2ECON *********************
C
C  PURPOSE: TO CONVERT AN ARRAY OF COLLISION  ENERGIES INTO A  SPECIFIED
C           FORM.
C
C  CALLING PROGRAM: GENERAL USE
C
C  SUBROUTINE:
C
C  INPUT :    (I*4)  INTYP   = 1 => 'EIN(ARRAY)' UNITS: DONOR   EV
C                              = 2 => 'EIN(ARRAY)' UNITS: RECVR   EV
C                              = 3 => 'EIN(ARRAY)' UNITS: ENERGY  EV/AMU
C  INPUT :    (I*4)  OUTTYP  = 1 => 'EOUT(ARRAY)' UNITS: DONOR  EV
C                              = 2 => 'EOUT(ARRAY)' UNITS: RECVR  EV
C                              = 3 => 'EOUT(ARRAY)' UNITS: ENERGY EV/AMU
C  INPUT :    (R*8)  AMD     = DONOR MASS NUMBER
C  INPUT :    (R*8)  AMR     = RECEIVER MASS NUMBER
C  INPUT :    (I*4)  IEVAL   = NO. OF ENERGIES IN EIN(ARRAY)
C  INPUT :    (R*8)  EIN()   = INPUT  ENERGIES (STATED UNITS)
C  OUTPUT:    (R*8)  EOUT()  = OUTPUT ENERGIES (STATED UNITS)
C
C
C             (I*4)  I       = GENERAL USE
C
C             (R*8)  ECONV() = ENERGY/VELOCITY CONVERSION PARAMETERS
C
C ROUTINES:  NONE
C
C
C AUTHOR:  H. P. SUMMERS, UNIVERSITY OF STRATHCLYDE
C          JA8.08
C          TEL.  0141-553-4196
C
C DATE:    17/11/95
C
C UNIX-IDL PORT: H.P.SUMMERS
C
C VERSION: 1.1 DATE: 19-11-96
C MODIFIED: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C     - PUT UNDER S.C.C.S. CONTROL
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IEVAL,       INTYP,       OUTTYP
      REAL*8              AMD,         AMR,         EIN(IEVAL)
      REAL*8              EOUT(IEVAL)
```

4.11 c2setp: Subroutine c2setp from library adas3xx

```
C
       SUBROUTINE C2SETP( NBSEL )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: C2SETP *********************
C
C  PURPOSE:  WRITES THE VALUE OF NBSEL OUT TO IDL
C
C  CALLING PROGRAM: ADAS302
C
C  DATA:
C           DATA IS OBTAINED VIA SUBROUTINE 'C2DATA'
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  NBSEL   = NUMBER OF ION/ATOM X-SECTIONS READ IN.
C                           I.E. NUMBER OF DATA-BLOCKS.
C
C          (I*4)  PIPEOU  = PARAMETER = UNIT NUMBER FOR OUTPUT TO PIPE
C
C ROUTINES:
C           ROUTINE     SOURCE      BRIEF DESCRIPTION
C           ------------------------------------------------------------
C
C
C AUTHOR:  H. P. SUMMERS, UNIVERSITY OF STRATHCLYDE
C          JA8.08
C          TEL. 0141-553-4196
C
C DATE:    12/11/96
C
C UNIX-IDL PORT: H.P.SUMMERS
C
C VERSION: 1.1 DATE: 19-11-96
C MODIFIED: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C     - PUT UNDER S.C.C.S. CONTROL
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      INTEGER             NBSEL
```

4.12 c2spln: Subroutine c2spln from library adas3xx

```
C
      SUBROUTINE C2SPLN( NEDIM  ,
     &                   IEA    , IEVAL   ,
     &                   TEEA   , EEVA    ,
     &                   SCX    , SCXA    ,
     &                   LERNG
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: C2SPLN *********************
C
C  PURPOSE:
C          PERFORMS CUBIC SPLINE ON LOG(ENERGY <EV/AMU> ) VERSUS
C          LOG(CX CROSS-SECTION <CM2>).
C          INPUT DATA FOR A GIVEN DONOR/RECEIVER COMBINATION DATA-BLOCK.
C
C          USING  ONE-WAY SPLINES  IT  CALCULATES  THE  CROSS-SECTIONS
C          FOR  'IEVAL'  COLLISION ENERGIES VALUES  FROM
C          THE LIST OF COLLISION ENERGIES READ IN FROM THE INPUT FILE
C
C          IF A  VALUE  CANNOT  BE  INTERPOLATED  USING  SPLINES  IT  IS
C          EXTRAPOLATED VIA 'XXSPLE'. (SEE NOTES BELOW).
C
C  CALLING PROGRAM: ADAS302/SSIA
C
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  IEA     = INPUT DATA FILE: NUMBER OF COLLISION ENER-
C                           GIES READ FOR THE DATA-BLOCK BEING ASSESSED
C  INPUT : (I*4)  IEVAL   = NUMBER OF ISPF ENTERED COLLISION ENERGIES
C                           VALUES FOR  WHICH  CX CROSS-SECTIONS
C                           ARE REQUIRED FOR TABULAR/GRAPHICAL OUTPUT.
C
C  INPUT : (R*8)  TEEA()  = INPUT DATA FILE: COLLISION ENERGIES (EV/AMU)
C                           FOR THE DATA-BLOCK BEING ASSESSED.
C                           DIMENSION: COLLISION ENERGY INDEX
C  INPUT : (R*8)  EEVA()  = USER ENTERED: COLLISION ENERGIES (EV/AMU)
C                           DIMENSION: COLLISION ENERGY INDEX
C
C  INPUT : (R*8)  SCX()    =INPUT DATA FILE: FULL SET OF CX CROSS-
C                           SECTIONS FOR THE DATA-BLOCK BEING ANALYSED
C                           1ST DIMENSION: COLLISION ENERGY INDEX
C  OUTPUT: (R*8)  SCXA()  = SPLINE INTERPOLATED OR  EXTRAPOLATED
C                           CX CROSS-SECTIONS FOR
C                           THE USER ENTERED COLLISION ENERGIES.
C                           DIMENSION: COLLISION ENERGIES INDEX
C
C  OUTPUT: (L*4)  LERNG()=  .TRUE.  => OUTPUT 'SCXA()' VALUE WAS INTER-
C                                      POLATED  FOR  THE  USER  ENTERED
C                                      COLLISION ENERGY 'EEVA()'.
C                           .FALSE. => OUTPUT 'SCXA()' VALUE WAS EXTRA-
C                                      POLATED  FOR  THE  USER  ENTERED
C                                      COLLISION ENERGY 'EEVA()'.
C                           DIMENSION: COLLISION ENERGY INDEX
C
C          (I*4)  NIN     = PARAMETER = MAX. NO. OF INPUT  ENERGY
C                                       VALUES. MUST BE >= 'IEA'
C          (I*4)  NOUT    = PARAMETER = MAX. NO. OF OUTPUT ENERGY
```

```
C                                       VALUES.  MUST BE >= 'IEVAL'
C          (I*4)  L1      = PARAMETER = 1
C
C
C          (I*4)  IET     = ARRAY SUBSCRIPT USED INPUT  FILE  COLLISION
C                           ENERGIES.
C          (I*4)  IT      = ARRAY  SUBSCRIPT  USED  FOR  USER  ENTERED
C                           COLLISION ENERGIES.
C          (I*4)  IOPT    = DEFINES THE BOUNDARY DERIVATIVES FOR THE
C                           SPLINE ROUTINE 'XXSPLE', SEE 'XXSPLE'.
C                           (VALID VALUES = <0, 0, 1, 2, 3, 4)
C
C          (L*4)  LSETX   = .TRUE.  => SET UP SPLINE PARAMETERS RELATING
C                                      TO 'XIN' AXIS.
C                           .FALSE. => DO NOT SET UP SPLINE PARAMETERS
C                                      RELATING TO 'XIN' AXIS.
C                                      (I.E. THEY WERE SET IN A PREVIOUS
C                                            CALL )
C                           (VALUE SET TO .FALSE. BY 'XXSPLE')
C
C          (R*8)  R8FUN1  = FUNCTION - (SEE ROUTINES SECTION BELOW)
C
C          (R*8)  XIN()   = LOG( DATA FILE COLLISION ENERGIES )
C          (R*8)  YIN()   = LOG( DATA FILE CX CROSS-SECTIONS)
C          (R*8)  XOUT()  = LOG( USER ENTERED COLLISION ENERGIES.)
C          (R*8)  YOUT()  = LOG( OUTPUT GENERATED CX CROSS-SECTIONS)
C          (R*8)  DF()    = SPLINE INTERPOLATED DERIVATIVES
C
C
C NOTE:
C
C          ONE-DIMENSIONAL SPLINE CARRIED OUT BY THIS SUBROUTINE:
C
C           LOG( SCX )  vs. LOG( E )
C
C           E      = COLLISION ENERGY (units: eV/AMU)
C           SCX    = CX CROSS-SECTION (units: cm**2)
C
C           Extrapolation criteria:
C
C           Low  E: zero gradient  extrapolation (i.e. DY(1)  = 0.0)
C           High E: zero curvature extrapolation (i.e. DDY(N) = 0.0)
C
C           (These criteria are met by calling XXSPLE with IOPT=4)
C
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          XXSPLE     ADAS      SPLINE SUBROUTINE (EXTENDED DIAGNOSTICS)
C          R8FUN1     ADAS      REAL*8 FUNCTION: ( X -> X )
C
C AUTHOR:  H. P. SUMMERS, UNIVERSITY OF STRATHCLYDE
C          JA8.08
C          TEL. 0141-553-4196
C
C DATE:    16/11/95
C
C UNIX-IDL PORT: H.P.SUMMERS
C
C VERSION: 1.1 DATE: 19-11-96
```

```
C MODIFIED: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C     - PUT UNDER S.C.C.S. CONTROL
C
C VERSION: 1.2 DATE: 27-10-2004
C MODIFIED: Martin O'Mullane
C     - Error message reported E9SPLN and not C2SPLN.
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IEA,         IEVAL,       NEDIM
      LOGICAL             LERNG(IEVAL)
      REAL*8              EEVA(IEVAL), SCX(NEDIM),  SCXA(IEVAL)
      REAL*8              TEEA(NEDIM)
```

4.13 c2vrdc: Subroutine c2vrdc from library adas3xx

```
C
      SUBROUTINE C2VRDC( ICIND  ,
     &                   NBSEL  ,
     &                   CDONOR , CRECVR, CFSTAT , CTYPE ,
     &                   SCI    ,
     &                   SDONOR , SRECVR, SFSTAT , STYPE
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: C2VRDC *********************
C
C  PURPOSE: TO SETUP DONOR/RECVR CHARACTER STRINGS FOR USE WITH SETTING
C           ISPF VARIABLES WHEN  SELECTED CROSS-SECTIONS ARE DISPLAYED.
C
C  CALLING PROGRAM: C2PAN2
C
C  SUBROUTINE:
C
C  INPUT : (I*4)   ICIND    = SELECTED/REQUESTED CROSS-SECTION DATA-BLOCK
C                             INDEX.
C
C  INPUT : (I*4)   NBSEL    = NUMBER OF DATA-BLOCK CROSS-SECTIONS.
C
C  INPUT : (C*9)   CDONOR() = INPUT DATA FILE: DONOR ION IDENTIFICATION
C                             DIMENSION: DATA-BLOCK INDEX
C  INPUT : (C*9)   CRECVR() = INPUT DATA FILE: RECEIVER ION IDENTIFICATION
C                             DIMENSION: DATA-BLOCK INDEX
C  INPUT : (C*10)  CFSTAT() = INPUT DATA FILE: FINAL STATE SPECIFICATION
C                             DIMENSION: DATA-BLOCK INDEX
C  INPUT : (C*2)   CTYPE()  = INPUT DATA FILE: CROSS-SECTION TYPE
C                             DIMENSION: DATA-BLOCK INDEX
C                             DIMENSION: DATA-BLOCK INDEX
C
C  OUTPUT: (C*3)   SCI      = ISPF PANEL VARIABLE: DATA-BLOCK INDEX
C
C  OUTPUT: (C*9)   SDONOR   = ISPF PANEL VARIABLE: DONOR ION IDENTIFICATION
C  OUTPUT: (C*9)   SRECVR   = ISPF PANEL VARIABLE: RECEIVER ION IDENTIFICATION
C  OUTPUT: (C*10)  SFSTAT   = ISPF PANEL VARIABLE: FINAL STATE SPECIFICATION
C  OUTPUT: (C*2)   STYPE    = ISPF PANEL VARIABLE: CROSS-SECTION TYPE
C
C
C ROUTINES: NONE
C
C AUTHOR:  H. P. SUMMERS, UNIVERSITY OF STRATHCLYDE
C          JA8.08
C          TEL.  0141-553-4196
C
C DATE:    15/11/95
C
C UNIX-IDL PORT: H.P.SUMMERS
C
C VERSION: 1.1 DATE: 19-11-96
C MODIFIED: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C     - PUT UNDER S.C.C.S. CONTROL
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
```

```
CHARACTER*9         CDONOR(NBSEL)
CHARACTER*10        CFSTAT(NBSEL)
CHARACTER*9         CRECVR(NBSEL)
CHARACTER*2         CTYPE(NBSEL)
CHARACTER*3         SCI
CHARACTER*9         SDONOR
CHARACTER*10        SFSTAT
CHARACTER*9         SRECVR
CHARACTER*2         STYPE
INTEGER             ICIND,       NBSEL
```

4.14 c2wr11: Subroutine c2wr11 from library adas3xx

```
C
       SUBROUTINE C2WR11( IUNIT    , UID    , DATE   ,
     &                    NSTORE   , NTDIM  ,
     &                    NBSEL    , ISELA  ,
     &                    CPRMYA   , CSCDYA ,
     &                    CTYPEA ,
     &                    DSFLLA   ,
     &                    AMDA     , AMRA   ,
     &                    ITA      ,
     &                    TPA      ,
     &                    QFTEQA   , QFTIAA
     &                  )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: C2WR11 *********************
C
C  PURPOSE:  TO  WRITE  DATA  TO THERMAL ION/ATOM
C            RATE COEFFICIENT PASSING FILE FOR GIVEN PRIMARY SPECIES.
C
C  CALLING PROGRAM: ADAS302
C
C  DATA:
C
C           UP TO 'NSTORE' SETS (DATA-BLOCKS) OF DATA MAY BE WRITTEN TO
C           THE FILE - EACH  BLOCK  FORMING  A  COMPLETE  SET OF  RATE-
C           COEFFICIENTS FOR THE  PRIMARY SPECEIS.
C
C           THE UNITS USED IN THE DATA FILE ARE TAKEN AS FOLLOWS:
C
C           TEMPERATURES        : EV
C           RATE COEFFICIENTS   : CM**3 SEC-1
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  IUNIT    = UNIT TO WHICH INPUT FILE IS ALLOCATED.
C  INPUT : (C*6)  UID      = USER IDENTIFIER OF PRODUCER
C  INPUT : (C*8)  DATE     = DATE
C  INPUT : (I*4)  NSTORE   = MAXIMUM NUMBER  OF DATA-BLOCKS THAT CAN
C                            BE WRITTEN.
C  INPUT : (I*4)  NTDIM    = MAX NUMBER OF RECEIVER & DONOR TEMPERATURES
C                            ALLOWED
C
C  INPUT : (I*4)  NBSEL    = NUMBER OF DATA-BLOCKS WRITTEN
C  INPUT : (I*4)  ISELA()  = WRITE - DATA-SET DATA-BLOCK ENTRY INDICES
C                            DIMENSION: DATA-BLOCK INDEX
C  INPUT : (C*5)  CPRMYA() = INPUT DATA FILE - SELECTED DATA-BLOCK:
C                            PRIMARY SPECEIS INDENTIFICATION
C                            DIMENSION: DATA-BLOCK INDEX
C  INPUT : (C*5)  CSCDYA() = INPUT DATA FILE - SELECTED DATA-BLOCK:
C                            SECONDARY SPECIES IDENTIFICATION
C                            DIMENSION: DATA-BLOCK INDEX
C  INPUT : (C*3)  CTYPEA() = INPUT DATA FILE - SELECTED DATA-BLOCK:
C                            CROSS-SECTION TYPE
C                            DIMENSION: DATA-BLOCK INDEX
C  INPUT : (R*8)  AMRA     = READ - SECONDARY SPECIES ATOMIC MASS
C                            DIMENSION: DATA-BLOCK INDEX
C  INPUT : (R*8)  AMDA     = READ - PRIMARY SPECIES  ATOMIC MASS
C                            DIMENSION: DATA-BLOCK INDEX
C  INPUT : (C*44) DSFLLA() = MVS DATA SET NAME OF SOURCE DATA SET
```

```
C                           DIMENSION: DATA-BLOCK INDEX
C
C
C  INPUT : (I*4)  ITA()    = READ - NUMBER OF TEMPERATURES
C                           DIMENSION: DATA-BLOCK INDEX
C
C  INPUT : (R*8)  TPA(,)    = READ - TEMPERATURES (UNITS: EV)
C                           1ST DIMENSION: TEMPERATURE INDEX
C                           2nd DIMENSION: DATA-BLOCK INDEX
C
C  INPUT : (R*8)  QFTEQA(,)= READ - EQUAL TEMPERATURE RATE-COEFFICIENTS
C                                  (UNITS: cm**3 sec-1)
C                           1st DIMENSION: SECONDARY TEMPERATURE INDEX
C                           2nd DIMENSION: DATA-BLOCK INDEX
C  INPUT : (R*8)  QFTIAA(,,)=READ - FULL SET OF RATE-COEFFICIENTS
C                                  (UNITS: cm**3 sec-1)
C                           1st DIMENSION: PRIMARY   TEMPERATURE INDEX
C                           2nd DIMENSION: SECONDARY TEMPERATURE INDEX
C                           3rd DIMENSION: DATA-BLOCK INDEX
C
C          (C*2)  CEQUAL    = PARAMETER = 'EQ'
C
C          (I*4)  I4EIZ0    = FUNCTION - (SEE ROUTINES SECTION BELOW)
C          (I*4)  I4FCTN    = FUNCTION - (SEE ROUTINES SECTION BELOW)
C          (I*4)  I4UNIT    = FUNCTION - (SEE ROUTINE SECTION BELOW)
C          (I*4)  IBLK      = ARRAY INDEX: DATA-BLOCK INDEX
C          (I*4)  ITR       = ARRAY INDEX: SECONDARY TEMPERATURE INDEX
C          (I*4)  ITD       = ARRAY INDEX: PRIMARY   TEMPERATURE INDEX
C          (I*4)  NTRNUM    = NUMBER OF SECONDARY TEMPERATURES FOR CURRENT
C                           DATA-BLOCK
C          (I*4)  NTDNUM    = NUMBER OF PRIMARY   TEMPERATURES FOR CURRENT
C                           DATA-BLOCK
C          (I*4)  IABT      = RETURN CODE FROM 'I4FCTN'
C
C
C          (C*10) IONNAM    = READ - PRIMARY SPECIES DESIGNATION STRING
C          (C*80) C80       = GENERAL USE 80 BYTE  CHARACTER  STRING  FOR
C                           THE INPUT OF DATA-SET RECORDS.
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4UNIT     ADAS      INTEGER*4 FUNCTION    -
C                               FETCH UNIT NUMBER FOR OUTPUT OF MESSAGES
C
C AUTHOR:  H. P. SUMMERS, UNIVERSITY OF STRATHCLYDE
C          JA8.08
C          TEL. 0141-553-4196
C
C DATE:    18/11/96
C
C UNIX-IDL PORT: H.P.SUMMERS
C
C VERSION: 1.1 DATE: 19-11-96
C MODIFIED: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C     - PUT UNDER S.C.C.S. CONTROL
C
C-----------------------------------------------------------------------
      CHARACTER*5         CPRMYA(NSTORE),           CSCDYA(NSTORE)
      CHARACTER*3         CTYPEA(NSTORE)
      CHARACTER*8         DATE
```

```
CHARACTER*80        DSFLLA(NSTORE)
CHARACTER*6         UID
INTEGER             ISELA(NSTORE),            ITA(NSTORE), IUNIT
INTEGER             NBSEL,       NSTORE,      NTDIM
REAL*8              AMDA(NSTORE),             AMRA(NSTORE)
REAL*8              QFTEQA(NTDIM,NSTORE)
REAL*8              QFTIAA(NTDIM,NTDIM,NSTORE)
REAL*8              TPA(NTDIM,NSTORE)
```

4.15 c3alrs: Subroutine c3alrs from library adas3xx

```
      SUBROUTINE C3ALRS( IORD , EM1  , EM2  ,
     &                   EPRO , TTAR , ETHR ,
     &                   YA   , N    , NS   ,
     &                   OA   , SQEF
     &                 )
C-----------------------------------------------------------------------
C
C  ********************** FORTRAN77 SUBROUTINE : C3ALRS ****************
C
C  PURPOSE:  COMPUTES ALPHAS AND REDUCED SPEEDS.  RETURNS AN EFFECTIVE
C            CHARGE-EXCHANGE RATE COEFFICIENT
C
C  CALLING PROGRAM: SQEF
C
C  SUBROUTINE:
C
C  INPUT : (I*4)   IORD  = 1 FOR 1ST PARTICLE INCIDENT AND MONOENERGETIC
C                        = 2 FOR 2ND PARTICLE INCIDENT AND MONOENERGETIC
C  INPUT : (I*4)   ISEL  = SELECTOR FOR PARTICULAR RATE COEFFT.
C                          CHOSEN FROM TABLE
C  INPUT : (R*8)   EM1   = ATOMIC MASS NUMBER FOR 1ST PARTICLE
C  INPUT : (R*8)   EM2   = ATOMIC MASS NUMBER FOR 2ND PARTICLE
C  INPUT : (R*8)   EPRO  = INCIDENT PARTICLE ENERGY (EV/AMU)
C  INPUT : (R*8)   TTAR  = MAXWELL TEMPERATURE OF TARGET PARTICLES (EV)
C  INPUT : (R*8)   ETHR  = THRESHOLD ENERGY
C  INPUT : (R*8)   ZSEL  = NUCLEAR CHARGE (REQUIRED ONLY
C                          FOR PARTICULAR ISEL)
C  INPUT : (I*4)   NSEL  = PRINC. QUANTUM NO. (REQUIRED ONLY
C                          FOR PARTICULAR ISEL
C                          NB. NSEL SHOULD BE ZERO ON ENTRY OTHERWISE)
C  INPUT : (I*4)   N     = NUMBER OF SOURCE DATA VALUES
C
C  OUTPUT: (R*8)   SQEF  = RATE COEFFICIENT (CM3 SEC-1)
C  OUTPUT: (R*8)   OA()  = RATE COEFFTS.(CM**3 SEC-1) FOR SELECTED
C                          SOURCE DATA
C  OUTPUT: (R*8)   EA()  = SET OF ENERGIES (EV/AMU) FOR
C                          SELECTED SOURCE DATA
C  I/O   : (R*8)   YA()  = ENERGIES ON INPUT, SPEEDS ON OUTPUT
C
C          (I*4)   ISWIT = ENERGY RANGE SWITCHING INDEX
C          (I*4)   I     = GENERAL INDEX
C          (I*4)   K     = GENERAL INDEX
C
C          (R*8)   ABI   = FUNCTION - SEE BELOW
C          (R*8)   EMT   = SELECTED MASS
C          (R*8)   F     = GENERAL VARIABLE
C          (R*8)   SUM   = GENERAL VARIABLE
C          (R*8)   SXI   = GENERAL VARIABLE
C          (R*8)   SXXI  = GENERAL VARIABLE
C          (R*8)   U     = GENERAL VARIABLE
C          (R*8)   V     = GENERAL VARIABLE
C          (R*8)   VTHR  = THRESHOLD SPEED
C          (R*8)   X     = GENERAL VARIABLE
C          (R*8)   XI    = GENERAL VARIABLE
C          (R*8)   XRMIN = GENERAL VARIABLE
C          (R*8)   XXI   = GENERAL VARIABLE
C          (R*8)   XA()  = GAUSS-LAGUERRE NODES   (9-POINT)
C          (R*8)   WXA() = GAUSS-LAGUERRE WEIGHTS (9-POINT)
C
```

```
C
C  ROUTINES:
C           ROUTINE    SOURCE    BRIEF DESCRIPTION
C           ------------------------------------------------------------
C           ABI        ADAS      COMPUTES INTEGRAL FOR RATE COEFFICIENT
C
C  AUTHOR:  C J. WHITEHEAD, PAP, UNIVERSITY OF STRATHCLYDE
C           EXT 4205
C
C  DATE:    14/11/94
C
C  UPDATE:  09/12/94 - HP SUMMERS: ADJUST FORMATTING
C
C  UPDATE:  03/05/95 - PE BRIDEN : ADD DATA DECLARATION FOR F AND EMT.
C                                  (STOPS COMPILATION WARNING OF
C                                   UNINITIALISED VARIABLES.)
C
C  UPDATE:  15/05/95 - TIM HAMMOND: UNIX PORT - PUT INTO SCCS
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      INTEGER             IORD,        N,           NS
      REAL*8              EM1,         EM2,         EPRO,        ETHR
      REAL*8              OA(24),      SQEF,        TTAR,        YA(24)
```

4.16 c3corr: Subroutine c3corr from library adas3xx

```
      SUBROUTINE C3CORR( NVALS   , IBSEL  ,
     &                   QATOM   , ATOM   ,
     &                   QEFREF  , ATMREF ,
     &                   NSTORE  , NA     ,
     &                   RION    , SCALED
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: C3CORR ********************
C
C  PURPOSE: APPLY A CORRECTION TO REFERENCE RATE COEFFICIENT TO ALLOW
C           FOR VARIATION OF PLASMA PARAMETERS ALONG ONE-DIMENSIONAL
C           SCANS
C
C  CALLING PROGRAM: SQEF
C
C  SUBROUTINE:
C
C  INPUT : (I*4)   NVALS()    = NUMBER OF VALUES IN SELECTED DATA BLOCK
C                               1ST. DIM: NSTORE
C  INPUT : (I*4)   IBSEL    = SELECTED BLOCK INDEX
C  INPUT : (R*8)   QATOM(,) = EFFECT. RATE COEFFT. TO SCALE WITH PARAM.
C                               1ST. DIM: NA
C                               2ND. DIM: NSTORE
C  INPUT : (R*8)   ATOM(,)  = PLASMA PARAM. VALUES TO BASE SCALING ON
C                               1ST. DIM: NA
C                               2ND. DIM: NSTORE
C  INPUT : (R*8)   QEFREF() = RATE COEFFICIENT REFERENCE VALUES
C                               1ST. DIM: NSTORE
C  INPUT : (R*8)   ATMREF() = REFERENCE VALUES FOR PLASMA PARAM.
C                               1ST. DIM: NSTORE
C  INPUT : (R*8)   RION     = SPECIFIED PARM. VALUE FOR COEFFT. OUTPUT
C  INPUT : (I*4)   NSTORE   = NUMBER OF DATA BLOCKS IN DATASET.
C  INPUT : (I*4)   NA       = MAX. NO. OF PARAM. VALUES IN BLOCK
C
C  OUTPUT: (R*8)   SCALED   = SCALED PLASMA PARAMETER
C
C  PARAM : (I*4)   MXIN     = MAX. NO. OF INPUT DATA SET VALUES
C
C          (I*4)   MAXVAL   = NUMBER OF DATA VALUES IN SELECTED SET
C          (I*4)   IOPT     = DEFINES THE BOUNDARY DERIVATIVES FOR THE
C                             SPLINE ROUTINE 'XXSPLE'. SEE 'XXSPLE'.
C                             ( VALID VALUES = <0, 0, 1, 2, 3, 4)
C          (I*4)   J        = LOOP INDEX.
C
C          (L*4)   LSETX    = FLAGS TO SPLINE ROUTINE 'XXSPLN' IF
C                             SPLINE PARAMETERS SHOULD BE SET UP.
C                             .TRUE.  => SET UP SPLINE PARAMS.
C                             .FALSE. => DO NOT SET UP SPLINE PARAMS.
C
C          (L*4)   LINTRP   = FLAGS TO IDENTIFY IF SCALED VALUE INTER-
C                             POLATED. (OUTPUT FROM XXSPLE).
C                             .TRUE.  => SCALED VALUE INTERPOLATED
C                             .FALSE. => SCALED VALUE EXTRAPOLATED.
C
C          (R*8)   VL()     = LN(INPUT VALUE/REF. VALUE) FOR PARAM.
C                             DIMENSION: 1
C          (R*8)   QVL()    = LN(INPUT TABLE RATE COEF.) FOR PARAM.
C                             DIMENSION: 1
```

```
C          (R*8)   VECL()    = LN(TABLE VALUE/REF. VALUE) FOR PARAM.
C                               DIMENSION: MXIN
C          (R*8)   QVECL()   = LN(TABLE RATE COEF./REF. RATE COEF.)
C                               DIMENSION: MXIN
C          (R*8)   DY()      = DERIVATIVES AT INPUT KNOTS TO XXSPLN.
C                               DIMENSION: MXIN
C
C  ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4UNIT     ADAS      RETURNS UNIT NO. FOR OUTPUT OF MESSAGES.
C          XXSPLE     ADAS      SPLINE SUBROUTINE (EXTENDED DIAGNOSTICS)
C          R8FUN1     ADAS      PERFORMS TRANSFORMATION ( X -> X )
C
C  NOTES:  THE QUANTITIES WHICH ARE SCALED BY THIS ROUTINE ARE :  ION
C          DENSITY, ION TEMPERATURE, EFFECTIVE Z AND MAGNETIC
C          FIELD THIS CODE IS TAKEN FROM THE OLDER ADAS CODE
C          'QEFFH.FOR' BY H.P.  SUMMERS
C
C  AUTHOR:  C.J. WHITEHEAD, PAP, UNIVERSITY OF STRATHCLYDE
C           EXT 4205
C
C  DATE:    24/11/94
C
C  UPDATE:  19/12/94 - HP SUMMERS: ADJUST FORMATTING
C
C  UPDATE:  21/04/95 - HP SUMMERS: REORDER TWO-DIMENSIONAL ARRAY INDICES
C
C  UPDATE:  03/05/95 - PE BRIDEN : 1) REPLACED CALLS TO SPLINE NAG
C                                      ROUTINES E01BAF/E02BBF WITH A
C                                      CALL TO XXSPLE. (REQUIRED SOME
C                                      RECODING). + CHECK FOR ATTEMPTED
C                                      EXTRAPOLATION.
C                                   2) MADE CHANGES TO CODE TO MAKE IT
C                                      ANSI STANDARD FORTRAN 77.
C                                   3) TIDIED UP CODE + ADDED CHECK TO
C                                      MAKE SURE INTERNAL ARRAYS ARE
C                                      LARGE ENOUGH.
C                                   4) GENERAL CHANGES TO FORMAT ETC.
C
C  UPDATE:  15/05/95 - Tim Hammond: UNIX PORT
C                      Put under SCCS control
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      INTEGER             IBSEL,       NA,          NSTORE
      INTEGER             NVALS(NSTORE)
      REAL*8              ATMREF(NSTORE),           ATOM(NA,NSTORE)
      REAL*8              QATOM(NA,NSTORE),         QEFREF(NSTORE)
      REAL*8              RION,        SCALED
```

4.17 c3data: Subroutine c3data from library adas3xx

```
      SUBROUTINE C3DATA( IUNIT  , DSNAME ,
     &                   NSTORE ,
     &                   NBSEL  , ISELA  ,
     &                   CWAVEL , CDONOR , CRECVR ,
     &                   CFILE  , CPCODE , CINDM  ,
     &                   QEFREF ,
     &                   ENREF  , TEREF  , DEREF  , ZEREF   , BMREF  ,
     &                   NENERA , NTEMPA , NDENSA , NZEFFA  , NBMAGA ,
     &                   ENERA  , TEMPA  , DENSA  , ZEFFA   , BMAGA  ,
     &                   QENERA , QTEMPA , QDENSA , QZEFFA  , QBMAGA
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: C3DATA *********************
C
C  PURPOSE : TO FETCH DATA FROM INPUT QEF DATA SET.
C
C  CALLING PROGRAM: ADAS303, SQEF
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  IUNIT     = UNIT NUMBER TO READ FROM
C  INPUT : (C*80) DSNAME    = MVS DATA SET NAME OF DATA SET BEING READ
C  INPUT : (I*4)  NSTORE    = MAXIMUM NUMBER OF DATA BLOCKS ALLOWED
C
C  OUTPUT: (I*4)  NBSEL     = NUMBER OF BLOCKS PRESENT
C  OUTPUT: (I*4)  ISELA()   = INDEX NUMBER OF DATA BLOCK
C
C  OUTPUT: (C*5)  CWAVEL()= INPUT DATA FILE: TRANSITION
C                           DIMENSION: DATA-BLOCK INDEX
C  OUTPUT: (C*8)  CDONOR()= INPUT DATA FILE: DONOR NEUTRAL ATOM
C                           DIMENSION: DATA-BLOCK INDEX
C  OUTPUT: (C*5)  CRECVR()= INPUT DATA FILE: RECEIVER NUCLEUS
C                           DIMENSION: DATA-BLOCK INDEX
C  OUTPUT: (C*8)  CFILE() = INPUT DATA FILE: SPECIFIC ION FILE SOURCE
C                           DIMENSION: DATA-BLOCK INDEX
C  OUTPUT: (C*8)  CPCODE()= INPUT DATA FILE: PROCESSING CODE
C                           DIMENSION: DATA-BLOCK INDEX
C  OUTPUT: (C*6)  CINDM() = FILE DATA FILE: EMISSION TYPE
C                           DIMENSION: DATA-BLOCK INDEX
C
C  OUTPUT: (R*8)  QEFREF()  = REFERENCE VALUE OF RATE COEFFICIENT
C  OUTPUT: (R*8)  ENREF()   =     "       "   "  ENERGY
C  OUTPUT: (R*8)  TEREF()   =     "       "   "  TEMPERATURE
C  OUTPUT: (R*8)  DEREF()   =     "       "   "  DENSITY
C  OUTPUT: (R*8)  ZEREF()   =     "       "   "  EFFECTIVE Z
C  OUTPUT: (R*8)  BMREF()   =     "       "   "  MAGNETIC FIELD
C  OUTPUT: (I*4)  NENERA()  = NUMBER OF ENERGIES
C  OUTPUT: (I*4)  NTEMPA()  = NUMBER OF TEMPERATURES
C  OUTPUT: (I*4)  NDENSA()  = NUMBER OF DENSITIES
C  OUTPUT: (I*4)  NZEFFA()  = NUMBER OF EFFECTIVE Z'S
C  OUTPUT: (I*4)  NBMAGA()  = NUMBER OF MAGNETIC FIELD VALUES
C                             1ST. DIM: NSTORE
C                             (FOR ABOVE ARRAYS)
C
C  OUTPUT: (R*8)  ENERA(,)  = ENERGIES
C  OUTPUT: (R*8)  QENERA(,) = RATE COEFFICIENTS FOR ENERGY VALUE
C  OUTPUT: (R*8)  TEMPA(,)  = TEMPERATURES
C  OUTPUT: (R*8)  QTEMPA(,) = RATE COEFFICIENTS FOR TEMPERATURES
```

```
C  OUTPUT: (R*8)  DENSA(,)  = DENSITIES
C  OUTPUT: (R*8)  QDENSA(,) = RATE COEFFICIENTS FOR DESNITIES
C  OUTPUT: (R*8)  ZEFFA(,)  = EFFECTIVE Z
C  OUTPUT: (R*8)  QZEFFA(,) = RATE COEFFICIENTS FOR EFFECTIVE Z
C  OUTPUT: (R*8)  BMAGA(,)  = MAGNETIC FIELD
C  OUTPUT: (R*8)  QBMAGA(,) = RATE COEFFICIENTS FOR MAGNETIC FIELDS
C                             1ST DIM: 12 OR 24  DEPENDING ON PARAMETER
C                             2ND DIM: NSTORE
C
C  ROUTINES:
C           ROUTINE    SOURCE    BRIEF DESCRIPTION
C           ------------------------------------------------------------
C           I4UNIT     ADAS      FETCHES FILE HANDLE FOR ERROR MESSAGE
C
C  AUTHOR:  H P SUMMERS, UNIVERSITY OF STRATHCLDYE
C           JA 8.08
C           TEL.  0141-553-4196
C
C  DATE:    19/04/95
C
C  UPDATE:  04/05/95 TIM HAMMOND - UNIX PORT
C                                  Increased DSNAME from 44 -> 80
C
C  UPDATE:  15/05/95 TIM HAMMOND - UNIX PORT
C                                  Changed delimiter character from
C    '\' to '!' as otherwise will not
C    compile.
C
C
C VERSION : 1.2
C DATE    : 02-12-2004
C MODIFIED: Martin O'Mullane
C        - Warn user that the routine is now deprecated
C                and that xxdata_12 should be used instead.
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      CHARACTER*8         CDONOR(NSTORE),           CFILE(NSTORE)
      CHARACTER*6         CINDM(NSTORE)
      CHARACTER*8         CPCODE(NSTORE)
      CHARACTER*5         CRECVR(NSTORE),           CWAVEL(NSTORE)
      CHARACTER*80        DSNAME
      INTEGER             ISELA(NSTORE),            IUNIT
      INTEGER             NBMAGA(NSTORE),           NBSEL
      INTEGER             NDENSA(NSTORE),           NENERA(NSTORE)
      INTEGER             NSTORE,      NTEMPA(NSTORE)
      INTEGER             NZEFFA(NSTORE)
      REAL*8              BMAGA(12,NSTORE),         BMREF(NSTORE)
      REAL*8              DENSA(24,NSTORE),         DEREF(NSTORE)
      REAL*8              ENERA(24,NSTORE),         ENREF(NSTORE)
      REAL*8              QBMAGA(12,NSTORE),        QDENSA(24,NSTORE)
      REAL*8              QEFREF(NSTORE),           QENERA(24,NSTORE)
      REAL*8              QTEMPA(12,NSTORE),        QZEFFA(12,NSTORE)
      REAL*8              TEMPA(12,NSTORE),         TEREF(NSTORE)
      REAL*8              ZEFFA(12,NSTORE),         ZEREF(NSTORE)
```

4.18 c3datao: Subroutine c3datao from library adas3xx

```
      SUBROUTINE C3DATAO( IUNIT  , IBSEL  ,
     &                    NBSEL  , NSTORE ,
     &                    IPASS  ,
     &                    IONNAM , QEFREF ,
     &                    TEREF  , DEREF  ,
     &                    ZEREF  , ENREF  ,
     &                    BMREF  , NENERA ,
     &                    NDENSA , NZEFFA ,
     &                    NBMAGA , NTEMPA ,
     &                    ENERA  , QENERA ,
     &                    TEMPA  , QTEMPA ,
     &                    DENSA  , QDENSA ,
     &                    ZEFFA  , QZEFFA ,
     &                    BMAGA  , QBMAGA )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: C3DATAO *******************
C
C ***********************************************************************
C * WARNING - WARNING - WARNING - WARNING - WARNING - WARNING - WARNING*
C *---------------------------------------------------------------------*
C * OLD VERSION OF C3DATA - ONLY USED BY SQEF (UNTIL SQEF UPDATED)     *
C ***********************************************************************
C
C  PURPOSE : READ IN VALUES FROM AN 'IONATOM' DATA SET OPENED BY
C            C3FILE
C
C  CALLING PROGRAM: SQEF
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  IUNIT     = UNIT NUMBER TO READ FROM
C                             (OPENED BY C3FILE)
C  INPUT : (I*4)  NSTORE    = ARRAY DIMENSION
C  INPUT : (I*4)  ISEL      = INDEX NUMBER OF SELECTED BLOCK
C                             FROM IONATOM FILE
C  INPUT : (I*4)  IPASS     = 0 IF DATA FILE TO BE READ IN AFRESH
C                           = 1 IF DATA FILE IS NOT TO BE READ IN AGAIN
C                             (IPASS IS SET TO 0 WHEN
C                              ISEL IS NOT A VALID INDEX)
C  OUTPUT: (I*4)  NBSEL     = NUMBER OF BLOCKS PRESENT
C  OUTPUT: (C*80) IONNAM    = NAME OF ION
C
C  OUTPUT: (R*8)  QEFREF()  = REFERENCE VALUE OF RATE COEFFICIENT
C  OUTPUT: (R*8)  ENREF()   =     "       "   "  ENERGY
C  OUTPUT: (R*8)  TEREF()   =     "       "   "  TEMPERATURE
C  OUTPUT: (R*8)  DEREF()   =     "       "   "  DENSITY
C  OUTPUT: (R*8)  ZEREF()   =     "       "   "  EFFECTIVE Z
C  OUTPUT: (R*8)  BMREF()   =     "       "   "  MAGNETIC FIELD
C  OUTPUT: (I*4)  NENERA()  = NUMBER OF ENERGIES
C  OUTPUT: (I*4)  NTEMPA()  = NUMBER OF TEMPERATURES
C  OUTPUT: (I*4)  NDENSA()  = NUMBER OF DENSITIES
C  OUTPUT: (I*4)  NZEFFA()  = NUMBER OF EFFECTIVE Z'S
C  OUTPUT: (I*4)  NBMAGA()  = NUMBER OF MAGNETIC FIELD VALUES
C                             1ST. DIM: NSTORE
C                             (FOR ABOVE ARRAYS)
C  OUTPUT: (R*8)  ENERA(,)  = ENERGIES
C  OUTPUT: (R*8)  QENERA(,) = RATE COEFFICIENTS FOR ENERGY VALUE
C  OUTPUT: (R*8)  TEMPA(,)  = TEMPERATURES
```

```
C  OUTPUT: (R*8)  QTEMPA(,) = RATE COEFFICIENTS FOR TEMPERATURES
C  OUTPUT: (R*8)  DENSA(,)  = DENSITIES
C  OUTPUT: (R*8)  QDENSA(,) = RATE COEFFICIENTS FOR DESNITIES
C  OUTPUT: (R*8)  ZEFFA(,)  = EFFECTIVE Z
C  OUTPUT: (R*8)  QZEFFA(,) = RATE COEFFICIENTS FOR EFFECTIVE Z
C  OUTPUT: (R*8)  BMAGA(,)  = MAGNETIC FIELD
C  OUTPUT: (R*8)  QBMAGA(,) = RATE COEFFICIENTS FOR MAGNETIC FIELDS
C                             1ST DIM: NSTORE
C                             2ND DIM: 12 OR 24  DEPENDING ON PARAMETER
C
C  ROUTINES:
C           ROUTINE    SOURCE    BRIEF DESCRIPTION
C           ------------------------------------------------------------
C           I4UNIT     ADAS      FETCHES FILE HANDLE FOR ERROR MESSAGE
C
C UPDATE: 15/05/95 -Tim Hammond UNIX PORT
C Put under SCCS control
C
C VERSION: 1.2 DATE: 08-11-99
C MODIFIED: RICHARD MARTIN
C CHANGED IONNAM*80(80) TO IONNAM(80)*80
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      CHARACTER*80        IONNAM(80)
      INTEGER             IBSEL,       IPASS,       IUNIT
      INTEGER             NBMAGA(NSTORE),           NBSEL
      INTEGER             NDENSA(NSTORE),           NENERA(NSTORE)
      INTEGER             NSTORE,      NTEMPA(NSTORE)
      INTEGER             NZEFFA(NSTORE)
      REAL*8              BMAGA(NSTORE,12),         BMREF(NSTORE)
      REAL*8              DENSA(NSTORE,24),         DEREF(NSTORE)
      REAL*8              ENERA(NSTORE,24),         ENREF(NSTORE)
      REAL*8              QBMAGA(NSTORE,12),        QDENSA(NSTORE,24)
      REAL*8              QEFREF(NSTORE),           QENERA(NSTORE,24)
      REAL*8              QTEMPA(NSTORE,12),        QZEFFA(NSTORE,12)
      REAL*8              TEMPA(NSTORE,12),         TEREF(NSTORE)
      REAL*8              ZEFFA(NSTORE,12),         ZEREF(NSTORE)
```

4.19 c3file: Subroutine c3file from library adas3xx

```
      SUBROUTINE C3FILE(IUNIT, ATNAME, IRCODE, DSNAME)
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: C3FILE *********************
C
C  PURPOSE: TO OPEN AN 'IONATOM' DATA SET FOR ATOM 'ATNAME'
C           CURRENTLY AVAILABLE ATOMS ARE : H, HE, LI
C
C
C
C  DATA SET OPENED: 'JETUID.<GROUP>.<TYPE>
C                           (<EXTENSION>#<ATOM SYMBOL>)'
C
C  CALLING PROGRAM: SQEF
C
C  SUBROUTINE:
C
C  INPUT : (I*4)   IUNIT   = UNIT TO WHICH DATA SET WILL BE CONNECTED
C          (C*2)   ATNAME  = NAME OF ATOM
C
C  OUTPUT: (I*4)   IRCODE  = RETURN CODE FROM SUBROUTINE:
C                            0 => DATA SET SUCCESSFULLY CONNECTED
C                            1 => REQUESTED DATA  SET  MEMBER  DOES  NOT
C                                 EXISTS - DATA SET NOT CONNECTED.
C                            9 => REQUESTED DATA  SET  EXISTS BUT CANNOT
C                                 BE OPENED.
C  OUTPUT: (C*44)  DSNAME  = FULL MVS NAME OF OPENED DATA SET
C
C          (I*4)   LENF1   = FIRST NON-BLANK CHR OF 'DSNAME' GROUP PART
C          (I*4)   LENF2   = LAST  NON-BLANK CHR OF 'DSNAME' GROUP PART
C          (I*4)   LENF3   = FIRST NON-BLANK CHR OF 'DSNAME' TYPE PART
C          (I*4)   LENF4   = LAST  NON-BLANK CHR OF 'DSNAME' TYPE PART
C          (I*4)   LENF5   = FIRST NON-BLANK CHR OF 'DSNAME' EXT PART
C          (I*4)   LENF6   = LAST  NON-BLANK CHR OF 'DSNAME' EXT PART
C          (I*4)   LENF7   = FIRST NON-BLANK CHR OF 'ATNAME'
C          (I*4)   LENF8   = LAST  NON-BLANK CHR OF 'ATNAME'
C
C          (C*6)   USERID  = ADAS SOURCE DATA USER ID
C          (C*8)   USRGRP  = ADAS SOURCE DATA GROUPNAME
C          (C*5)   USRTYP  = ADAS SOURCE DATA TYPENAME
C          (C*4)   USREXT  = ADAS SOURCE DATA EXTENSION
C
C          (L*4)   LEXIST  = .TRUE.  => REQUESTED  DATA  SET  EXISTS.
C                            .FALSE. => REQUESTED  DATA  SET  DOES  NOT
C                                       EXIST.
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          XXUID      ADAS      FETCHES/SETS ADAS SOURCE DATA USER ID
C          XXSQEF     ADAS      FETCHES/SETS ADAS SOURCE DATA FILENAME
C          XXSLEN     ADAS      OBTAINS FIRST AND LAST NON-BLANK
C                               CHARACTERS IN A STRING
C
C AUTHOR:  C.J. WHITEHEAD, STRATHCLYDE
C          EXT 4205
C          BASED ON CODE IN F1FILE.F
C
C
```

```
C DATE:    24/11/94
C
C UPDATE:  15/05/95 - Tim Hammond - UNIX PORT
C                                   Added SCCS Header
C
C VERSION: 1.2 DATE: 08-11-99
C MODIFIED: RICHARD MARTIN
C REMOVED ACTION KEYWORD FROM OPEN STATEMENT.
C
C-------------------------------------------------------------------------
C-------------------------------------------------------------------------
C
C-------------------------------------------------------------------------
      CHARACTER*2         ATNAME
      CHARACTER*44        DSNAME
      INTEGER             IRCODE,      IUNIT
```

4.20 c4data: Subroutine c4data from library adas3xx

```
      SUBROUTINE C4DATA( IUNIT  , MXBE   , MXTD   , MXTT   ,
     &                   ITZ    , TSYM   , BEREF  , TDREF  ,
     &                   TTREF  , SVREF  , NBE    , BE     ,
     &                   NTDENS , TDENS  , NTTEMP , TTEMP  ,
     &                   SVT    , SVED   , DSNIN
     &                 )
C
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: C4DATA *********************
C
C  PURPOSE: TO READ DATA FROM AN EFFECTIVE BEAM STOPPING DATA SET.
C           (ADAS FORMAT ADF21).
C
C  CALLING PROGRAM: SBMS / ADAS304
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  IUNIT     = UNIT TO WHICH DATA SET IS CONNECTED.
C  INPUT : (I*4)  MXBE      = MAXIMUM NUMBER OF BEAM ENERGIES WHICH CAN
C                             BE READ.
C  INPUT : (I*4)  MXTD      = MAXIMUM NUMBER OF TARGET DENSITIES WHICH
C                             CAN BE READ.
C  INPUT : (I*4)  MXTT      = MAXIMUM NUMBER OF TARGET TEMPERATURES
C                             WHICH CAN BE READ.
C  INPUT : (C*80) DSNIN     = NAME OF FILE TO BE READ.
C  OUTPUT: (I*4)  ITZ       = TARGET ION CHARGE.
C  OUTPUT: (C*2)  TSYM      = TARGET ION ELEMENT SYMBOL.
C  OUTPUT: (R*8)  BEREF     = REFERENCE BEAM ENERGY.
C                             UNITS: EV/AMU
C  OUTPUT: (R*8)  TDREF     = REFERENCE TARGET DENSITY.
C                             UNITS: CM-3
C  OUTPUT: (R*8)  TTREF     = REFERENCE TARGET TEMPERATURE.
C                             UNITS: EV
C  OUTPUT: (R*8)  SVREF     = STOPPING COEFFT. AT REFERENCE BEAM ENERGY,
C                             TARGET DENSITY AND TEMPERATURE.
C                             UNITS: CM3 S-1
C  OUTPUT: (I*4)  NBE       = NUMBER OF BEAM ENERGIES.
C  OUTPUT: (R*8)  BE()      = BEAM ENERGIES.
C                             UNITS: EV/AMU
C                             DIMENSION: MXBE
C  OUTPUT: (I*4)  NTDENS    = NUMBER OF TARGET DENSITIES.
C  OUTPUT: (R*8)  TDENS()   = TARGET DENSITIES.
C                             UNITS: CM-3
C                             DIMENSION: MXTD
C  OUTPUT: (I*4)  NTTEMP    = NUMBER OF TARGET TEMPERATURES.
C  OUTPUT: (R*8)  TTEMP()   = TARGET TEMPERATURES.
C                             UNITS: EV
C                             DIMENSION: MXTT
C  OUTPUT: (R*8)  SVT()     = STOPPING COEFFT. AT REFERENCE BEAM ENERGY
C                             AND TARGET DENSITY.
C                             UNITS: CM3 S-1
C                             DIMENSION: MXTT
C  OUTPUT: (R*8)  SVED(,)   = STOPPING COEFFT. AT REFERENCE TARGET
C                             TEMPERATURE.
C                             UNITS: CM3 S-1
C                             1ST DIMENSION: MXBE
C                             2ND DIMENSION: MXTD
```

```
C
C          (I*4)  I         = ARRAY / LOOP INDEX.
C          (I*4)  J         = ARRAY INDEX.
C
C          (C*80) LINE      = TEXT LINE IN DATA SET.
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4UNIT     ADAS      RETURNS UNIT NO. FOR OUTPUT OF MESSAGES.
C
C AUTHOR:  JONATHAN NASH  (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/87
C          JET EXT. 5183
C
C DATE:    07/12/93
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 16-11-95
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST VERSION
C
C VERSION: 1.2                          DATE: 06-06-2003
C MODIFIED: Martin O'Mullane
C                   - Warn user that the routine is now deprecated
C                     and that xxdata_21 should be used instead.
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      CHARACTER*80        DSNIN
      CHARACTER*2         TSYM
      INTEGER             ITZ,          IUNIT,       MXBE,        MXTD
      INTEGER             MXTT,         NBE,         NTDENS,      NTTEMP
      REAL*8              BE(MXBE),     BEREF,       SVED(MXBE,MXTD)
      REAL*8              SVREF,        SVT(MXTT),   TDENS(MXTD), TDREF
      REAL*8              TTEMP(MXTT), TTREF
```

4.21 c4mnmx: Subroutine c4mnmx from library adas3xx

```
      SUBROUTINE C4MNMX( NSITYP , MXA , NA , A , AMNMX  )
C
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: C4MNMX *********************
C
C  PURPOSE: GIVEN A 2D ARRAY 'A' IT RETURNS THE MINIMUM AND MAXIMUM OF
C           BOTH 'A(1,I)' AND 'A(NA(I),I)'
C
C           WHERE I=1,NSITYP. ITS MAIN USE IS TO FIND THE RANGE ON
C           THE MINIMUM AND MAXIMUM OF THE BEAM ENERGIES, AND TARGET
C           DENSITIES AND TEMPERATURES ACROSS THE DIFFERENT ION TYPES.
C
C  CALLING PROGRAM: ADAS304
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  NSITYP    = NUMBER OF DIFFERENT ION TYPES.
C  INPUT : (I*4)  MXA       = SIZE OF FIRST DIMENSION OF 'A(,)'.
C  INPUT : (I*4)  NA()      = NO. OF ENTRIES IN EACH COLUMN OF 'A(,)'.
C                             DIMENSION: NSITYP
C  INPUT : (R*8)  A(,)      = INPUT ARRAY.
C                             1ST DIMENSION: MXA
C                             2ND DIMENSION: NSITYP
C
C  OUTPUT: (R*8)  AMNMX()   = MINIMUM/MAXIMUM VALUES.
C                             INDICES ARE AS FOLLOWS:
C                             1 => MIN OF 'A(1,I)
C                             2 => MAX OF 'A(1,I)
C                             3 => MIN OF 'A(NA(I),I)
C                             4 => MAX OF 'A(NA(I),I)
C                                  WHERE I=1,NSITYP.
C                             DIMENSION: 4
C
C           (I*4)  I         = LOOP INDEX.
C           (I*4)  J         = ARRAY INDEX.
C
C ROUTINES: NONE
C
C AUTHOR:  JONATHAN NASH  (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/87
C          JET EXT. 5183
C
C DATE:    17/12/93
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 16-11-95
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST VERSION
C
C VERSION: 1.2                          DATE: 17-05-07
C MODIFIED: Allan Whiteford
C               - Updated comments as part of subroutine documentation
C                 procedure.
C
C-----------------------------------------------------------------------
C
```

```
C-----------------------------------------------------------------------
      INTEGER             MXA,         NA(NSITYP),  NSITYP
      REAL*8              A(MXA,NSITYP),            AMNMX(4)
```

4.22 c4spln: Subroutine c4spln from library adas3xx

```
      SUBROUTINE C4SPLN( MXBE   , MXTD   , MXTT   , MXREQ  ,
     &                   NREQ   , BMENGA , DENSA  , TIA    ,
     &                   NSITYP , SVREF  , NBE    , BE     ,
     &                   NTDENS , TDENS  , NTTEMP , TTEMP  ,
     &                   SVT    , SVED   , SVREQ  , LIBMA  ,
     &                   LIDNA  , LITIA  , ZEFF   , ITZ    ,
     &                   LSET   )
C
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: C4SPLN *********************
C
C  PURPOSE: CALCULATES THE BEAM STOPPING COEFFICIENT FOR EACH TRIPLET
C           OF BEAM ENERGY, ION DENSITY AND ION TEMPERATURE. IT USES
C           A ONE-WAY CUBIC SPLINE FOR THE TEMPERATURE AND A TWO-WAY
C           CUBIC SPLINE FOR THE ENERGY/DENSITY PAIR TO DETERMINE THE
C           STOPPING COEFFICIENT FROM THE INPUT DATA SET. IF A VALUE
C           CANNOT BE INTERPOLATED USING SLPINES THEN IT IS
C           EXTRAPOLATED BY 'XXSPLE'.
C
C  CALLING PROGRAM: CXBMS / ADAS304
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  MXBE      = MAXIMUM NUMBER OF BEAM ENERGIES WHICH CAN
C                             BE READ.
C  INPUT : (I*4)  MXTD      = MAXIMUM NUMBER OF TARGET DENSITIES WHICH
C                             CAN BE READ.
C  INPUT : (I*4)  MXTT      = MAXIMUM NUMBER OF TARGET TEMPERATURES
C                             WHICH CAN BE READ.
C  INPUT : (I*4)  MXREQ     = MAXIMUM NUMBER OF REQUESTED TRIPLETS OF
C                             BEAM ENERGY, ION DENSITY AND ION TEMP.
C  INPUT : (I*4)  NREQ      = NUMBER OF REQUESTED TRIPLETS OF BEAM
C                             ENERGY, ION DENSITY AND ION TEMP.
C  INPUT : (R*8)  BMENGA()  = REQUESTED BEAM ENERGIES.
C                             UNITS: EV/AMU
C                             DIMENSION: NREQ
C  INPUT : (R*8)  DENSA()   = REQUESTED ION DENSITIES.
C                             UNITS: CM-3
C                             DIMENSION: NREQ
C  INPUT : (R*8)  TIA()     = REQUESTED ION TEMPERATURES.
C                             UNITS: EV
C                             DIMENSION: NREQ
C  INPUT : (I*4)  NSITYP    = NUMBER OF STOPPING ION TYPES.
C  INPUT : (R*8)  SVREF()   = STOPPING COEFFT. AT REFERENCE BEAM ENERGY,
C                             TARGET DENSITY AND TEMPERATURE.
C                             UNITS: CM3 S-1
C                             DIMENSION: NSITYP
C  INPUT : (I*4)  NBE()     = NUMBER OF BEAM ENERGIES.
C                             DIMENSION: NSITYP
C  INPUT : (R*8)  BE(,)     = BEAM ENERGIES.
C                             UNITS: EV/AMU
C                             1ST DIMENSION: MXBE
C                             2ND DIMENSION: NSITYP
C  INPUT : (I*4)  NTDENS()  = NUMBER OF TARGET DENSITIES.
C                             DIMENSION: NSITYP
C  INPUT : (R*8)  TDENS(,)  = TARGET DENSITIES.
C                             UNITS: CM-3
```

```
C                               1ST DIMENSION: MXTD
C                               2ND DIMENSION: NSITYP
C  INPUT : (I*4)  NTTEMP()   = NUMBER OF TARGET TEMPERATURES.
C                               DIMENSION: NSITYP
C  INPUT : (R*8)  TTEMP(,)    = TARGET TEMPERATURES.
C                               UNITS: EV
C                               1ST DIMENSION: MXTT
C                               2ND DIMENSION: NSITYP
C  INPUT : (R*8)  SVT(,)      = STOPPING COEFFT. AT REFERENCE BEAM ENERGY
C                               AND TARGET DENSITY.
C                               UNITS: CM3 S-1
C                               1ST DIMENSION: MXTT
C                               2ND DIMENSION: NSITYP
C  INPUT : (R*8)  SVED(,,)    = STOPPING COEFFT. AT REFERENCE TARGET
C                               TEMPERATURE.
C                               UNITS: CM3 S-1
C                               1ST DIMENSION: MXBE
C                               2ND DIMENSION: MXTD
C                               3RD DIMENSION: NSITYP
C  OUTPUT: (R*8)  SVREQ(,)    = STOPPING COEFFT. AT REQUESTED BEAM ENERGY,
C                               ION DENSITY AND ION TEMPERATURE.
C                               UNITS: CM3 S-1
C                               1ST DIMENSION: MXREQ
C                               2ND DIMENSION: NSITYP
C  OUTPUT: (L*4)  LIBMA(,)    = FLAGS IF INTERPOLATION OR EXTRAPOLATION
C                               USED FOR REQUESTED BEAM ENERGIES.
C                               .TRUE.  => INTERPOLATION USED.
C                               .FALSE. => EXTRAPOLATION USED.
C                               1ST DIMENSION: MXREQ
C                               2ND DIMENSION: NSITYP
C  OUTPUT: (L*4)  LIDNA(,)    = FLAGS IF INTERPOLATION OR EXTRAPOLATION
C                               USED FOR REQUESTED ION DENSITIES.
C                               .TRUE.  => INTERPOLATION USED.
C                               .FALSE. => EXTRAPOLATION USED.
C                               1ST DIMENSION: MXREQ
C                               2ND DIMENSION: NSITYP
C  OUTPUT: (L*4)  LITIA(,)    = FLAGS IF INTERPOLATION OR EXTRAPOLATION
C                               USED FOR REQUESTED ION TEMPERATURES.
C                               .TRUE.  => INTERPOLATION USED.
C                               .FALSE. => EXTRAPOLATION USED.
C                               1ST DIMENSION: MXREQ
C                               2ND DIMENSION: NSITYP
C  INPUT:  (R*8)  ZEFF()      = USED AS A WEIGHTING FACTOR ASSOCIATED
C                               WITH THE EVALUATION OF AN EFFECTIVE
C                               DENSITY.
C  INPUT:  (I*4)  ITZ()       = ARRAY CONTAINING THE NUCLEAR CHARGE OF
C       EACH IMPURITY CONSIDERED.
C  INPUT:  (L)    LSET        = LOGICAL FLAGGING WHETHER OR NOT THE INPUT
C                               DATASET VECTOR HAS CHANGED. IF SO, A
C                               REQUEST TO REDO THE SPLINES IS PASSED TO
C                               'XXSPLF'.
C
C  PARAM : (I*4)  MXI         = MAX. NO. OF STOPPING ION TYPES >= NSITYP.
C  PARAM : (I*4)  MXIN        = MAX. NO. OF INPUT DATA SET VALUES
C                               >= MXBE , MXTD , MXTT.
C  PARAM : (I*4)  MXOUT       = MAX. NO. OF OUTPUT VALUES >= NREQ.
C
C          (I*4)  IOPT        = DEFINES THE BOUNDARY DERIVATIVES FOR THE
C                               SPLINE ROUTINE 'XXSPLE'. SEE 'XXSPLE'.
C                               ( VALID VALUES = <0, 0, 1, 2, 3, 4)
C          (I*4)  NOUT        = NUMBER OF OUTPUT VALUES FOR SPLINE.
```

```
C          (I*4)  I         = LOOP INDEX.
C          (I*4)  J         = LOOP INDEX.
C          (I*4)  K         = LOOP INDEX.
C
C          (L*4)  LSETX     = FLAGS TO SPLINE ROUTINE 'XXSPLF' IF
C                             'X' SPLINE PARAMETERS SHOULD BE SET UP.
C                             .TRUE.  => SET UP SPLINE PARAMS.
C                             .FALSE. => DO NOT SET UP SPLINE PARAMS.
C          (L*4)  LSETY     = FLAGS TO SPLINE ROUTINE 'XXSPLE' IF
C                             'Y' SPLINE PARAMETERS SHOULD BE SET UP.
C                             .TRUE.  => SET UP SPLINE PARAMS.
C                             .FALSE. => DO NOT SET UP SPLINE PARAMS.
C
C          (R*8)  DYT(,)    = DERIVATIVES FOR SPLINE INTERPOLATION OVER
C                             TEMPERATURE. ONE VECTOR FOR EACH TARGET
C                             ION.  SAVED FOR SPEED ON MULTIPLE CALLS
C                             DIMENSION: (MXIN,MXI)
C          (R*8)  QT(,)     = SPLINE INTERPOLATED SECOND DERIVATIVES.
C          (R*8)  D1T(,)    = MULTIPLICATION FACTOR USED IN XXSPLF.
C          (R*8)  D2T(,)    = MULTIPLICATION FACTOR USED IN XXSPLF.
C          (R*8)  D3T(,)    = MULTIPLICATION FACTOR USED IN XXSPLF.
C          (R*8)  DYE(,,)   = DERIVATIVES FOR SPLINE INTERPOLATION OVER
C                             ENERGY. ONE VECTOR FOR EACH INPUT ENERGY
C                             AND TARGET ION.  SAVED FOR SPEED ON
C                             MULTIPLE CALLS
C                             DIMENSION: (MXIN,MXIN,MXI)
C          (R*8)  QE(,)     = SPLINE INTERPOLATED SECOND DERIVATIVES.
C          (R*8)  D1E(,)    = MULTIPLICATION FACTOR USED IN XXSPLF.
C          (R*8)  D2E(,)    = MULTIPLICATION FACTOR USED IN XXSPLF.
C          (R*8)  D3E(,)    = MULTIPLICATION FACTOR USED IN XXSPLF.
C          (R*8)  DYD(,)    = DERIVATIVES FOR SPLINE INTERPOLATION OVER
C                             DENSITY. ONE VECTOR FOR EACH TARGET ION.
C                             SAVED FOR SPEED ON MULTIPLE CALLS
C                             DIMENSION: (MXIN,MXI)
C          (R*8)  QD(,)     = SPLINE INTERPOLATED SECOND DERIVATIVES.
C          (R*8)  D1D(,)    = MULTIPLICATION FACTOR USED IN XXSPLF.
C          (R*8)  D2D(,)    = MULTIPLICATION FACTOR USED IN XXSPLF.
C          (R*8)  D3D(,)    = MULTIPLICATION FACTOR USED IN XXSPLF.
C
C          (R*8)  YOUT()    = Y OUTPUT ARRAY FROM SPLINE ROUTINE.
C                             DIMENSION: MXOUT
C
C          (R*8)  SVTO(,)   = STOPPING COEFFICIENTS AT REQUESTED ION
C                             TEMPERATURES.
C                             1ST DIMENSION: MXOUT
C                             2ND DIMENSION: MXI
C          (R*8)  SVEDO(,)  = STOPPING COEFFICIENTS AT REQUESTED BEAM
C                             ENERGIES AND ION DENSITY.
C                             1ST DIMENSION: MXOUT
C                             2ND DIMENSION: MXI
C          (R*8)  YPASS(,)  = STOPPING COEFFICIENTS AT REQUESTED BEAM
C                             ENERGIES.
C                             1ST DIMENSION: MXIN
C                             2ND DIMENSION: MXOUT
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4UNIT     ADAS      RETURNS UNIT NO. FOR OUTPUT OF MESSAGES.
C          XXSPLF     ADAS      SPLINE SUBROUTINE (EXTENDED DIAGNOSTICS)
C          R8FUN1     ADAS      PERFORMS TRANSFORMATION ( X -> X )
```

```
C
C AUTHOR:  JONATHAN NASH  (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/87
C          JET EXT. 5183
C
C DATE:    10/12/93
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 16-11-95
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST VERSION
C VERSION: 1.2
C MODIFIED: HARVEY ANDERSON
C    - THE BEAM STOPPING COEFFICIENT FOR EACH
C   INDIVIDUAL IMPURITY WAS BEING EVALUATED
C   AT THE WRONG DENSITY. THE BEAM STOPPING
C   COEFFICIENT SHOULD BE EVALUATED AT AN
C   EFFECTIVE DENSITY. THIS HAS BEEN CORRECTED.
C - THE TARGET DENSITY READ FROM ADF21 FILE IS THE
C   ELECTRON DENSITY. THE CORRECTION TO THE EVALUATION
C   OF THE EFFECTIVE DENSITY WAS DONE IN TERMS OF THE
C   ION DENSITY. THIS WAS CORRECTED SO THAT THE EFFECTIVE
C   DENSITY IS EVALUATED IN TERMS OF THE ELECTRON DENSITY.
C   20/12/96
C - INTRODUCED THE PARAMETER FACT2, TO ENABLE THE
C   EFFECTIVE ELECTRON DENSITY TO BE EVALUATED.ORIGINALY
C   THE USER WOULD ENTER THE TOTAL ION DENSITY AND
C   THE STOPPING COEFFICIENTS WOULD BE EVALUATED AT AN
C   EFFECTIVE ELECTRON DENSITY. NOW THE CODE HAS BEEN
C   CHANGED TO ALLOW THE USER TO ENTER THE TOTAL ELECTRON
C   DENSITY.
C
C VERSION: 1.3                          DATE: 19-03-03
C MODIFIED: LORNE HORTON
C               - INCREASED MXOUT TO ALLOW UP TO 1024 EVALUATIONS PER
C                 CALL.
C               - IMPLEMENTED XXSPLF TO SPEED BICUBIC SPLINING BY
C                 HOLDING AS MUCH AS POSSIBLE IN GLOBAL VARIABLES.
C               - REPLACED FACT1 AND FACT2 WITH ZEFF
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             ITZ(NSITYP), MXBE,        MXREQ,       MXTD
      INTEGER             MXTT,        NBE(NSITYP), NREQ,        NSITYP
      INTEGER             NTDENS(NSITYP),           NTTEMP(NSITYP)
      LOGICAL             LIBMA(MXREQ,NSITYP),      LIDNA(MXREQ,NSITYP)
      LOGICAL             LITIA(MXREQ,NSITYP),      LSET
      REAL*8              BE(MXBE,NSITYP),          BMENGA(NREQ)
      REAL*8              DENSA(NREQ), SVED(MXBE,MXTD,NSITYP)
      REAL*8              SVREF(NSITYP),            SVREQ(MXREQ,NSITYP)
      REAL*8              SVT(MXTT,NSITYP),         TDENS(MXTD,NSITYP)
      REAL*8              TIA(NREQ),   TTEMP(MXTT,NSITYP)
      REAL*8              ZEFF(NREQ)
```

4.23 c5dplr: Subroutine c5dplr from library adas3xx

```
       subroutine c5dplr( ndpix  , npix   , wvmin  , wvmax  ,
     &                    ndcomp , ncomp  , wvcomp , emcomp ,
     &                    tev    , amss   ,
     &                    total
     &                  )

C-----------------------------------------------------------------------
C
C  ****************** fortran77 subroutine: c5pixv *********************
C
C  Purpose:  Distribute Doppler broadened line emission into pixel range
C
C  Calling program: adas305, stark
C
C
C  Subroutine:
C
C  input : (i*4)  ndpix   = maximum number of pixels
C  input : (i*4)  npix    = number of pixels assigned to wavelength interval
C  input : (r*8)  wvmin   = lower limit of wavelength interval (ang)
C  input : (r*8)  wvmax   = upper limit of wavelength interval (ang)
C
C  input : (i*4)  ndcomp  = maximum number of components in feature
C  input : (i*4)  ncomp   = number of components in feature
C  input : (r*8)  wvcomp  = wavelenghts of components (ang)
C  input : (r*8)  emcomp  = emissivity of component
C
C  input : (r*8)  tev     = electron temperature (eV)
C  input : (r*8)  amss    = atomic mass number
C
C  output: (r*8)  total() = Doppler broadened emission in wavelength interval
C
C Routines:
C          Routine    Source    Brief Description
C          ------------------------------------------------------------
C
C
C Author:  Martin O'Mullane
C Date:    18-02-2005
C
C
C VERSION  : 1.1
C DATE     : 18-02-2005
C MODIFIED : Martin O'Mullane
C              - First version.
C
C-----------------------------------------------------------------------
      INTEGER             NCOMP,       NDCOMP,      NDPIX,       NPIX
      REAL*8              AMSS,        EMCOMP(NDCOMP),           TEV
      REAL*8              TOTAL(NDPIX),             WVCOMP(NDCOMP)
      REAL*8              WVMAX,       WVMIN
```

4.24 c5pixv: Subroutine c5pixv from library adas3xx

```
       subroutine c5pixv( ndpix  , npix   , wvmin , wvmax ,
     &                    cpixmx ,
     &                    tev    , amssno , wvl   , pec   ,
     &                    cpixa  , ind1   , ind2
     &                  )

C-----------------------------------------------------------------------
C
C  ****************** fortran77 subroutine: c5pixv *********************
C
C  Purpose:  Distribute Doppler broadened line emission into pixel range
C
C  Calling program: adas305, stark
C
C
C  Subroutine:
C
C  input : (i*4)  ndpix   = maximum number of pixels
C
C  input : (i*4)  npix    = number of pixels assigned to wavelength interval
C  input : (r*8)  wvmin   = lower limit of wavelength interval (ang)
C  input : (r*8)  wvmax   = upper limit of wavelength interval (ang)
C
C  input : (r*8)  cpixmx  = largest pixel count currently found
C                           for the wavelength range
C
C  input : (r*8)  tev     = electron temperature (eV)
C  input : (r*8)  amssno  = atomic mass number
C  input : (r*8)  wvl     = input line wavelength for test(ang)
C  input : (r*8)  pec     = emissivity coefficient for component
C
C  output: (r*8)  cpixa() = counts in each pixel for the line
C  output: (r*8)  ind1    = first pixel with non-negligible count
C  output: (r*8)  ind2    = last pixel with non-negligible count
C
C          (r*8)  fcrit   = pixel counts for the selected line below
C                           this fraction of the largest pixel count are
C                           discounted.
C Routines:
C          Routine    Source    Brief Description
C          ------------------------------------------------------------
C          r8erfc     ADAS      returns erfc(x) function value
C
C
C Author:  Martin O'Mullane
C Date:    18-02-2005
C
C Notes:   Based on hapixv.for in adas810.
C
C
C
C VERSION  : 1.1
C DATE     : 18-02-2005
C MODIFIED : Martin O'Mullane
C              - First version.
C
C VERSION  : 1.2
C DATE     : 04-07-2007
C MODIFIED : Hugh Summers
```

```
C             - Corrected error in ind1 & ind2 return (see xxpixv).
C
C-----------------------------------------------------------------------
      INTEGER             IND1,         IND2,         NDPIX,        NPIX
      REAL*8              AMSSNO,       CPIXA(NDPIX),               CPIXMX
      REAL*8              PEC,          TEV,          WVL,          WVMAX
      REAL*8              WVMIN
```

4.25 c5rlsp: Subroutine c5rlsp from library adas3xx

```
       subroutine c5rlsp( xn  , xl  , xml  , xms  ,
     &                    xn1 , xl1 , xml1 , xms1 ,
     &                    er  , ei
     &                  )

C-----------------------------------------------------------------------
C
C  ****************** fortran77 subroutine: c5rlsp *********************
C
C  PURPOSE:  Evaluates relativistic+spin orbit matrix elements  of the
C            form <phi|H|phi_1> for hydrogen in the nlm_lm_s individual
C            set basis.
C
C  CALLING PROGRAM: stark (adas305)
C
C  SUBROUTINE:
C
C  INPUT : (R*8)   xn    = principal quantum number (bra state)
C  INPUT : (R*8)   xl    = orbital angular momentum quantum number
C  INPUT : (R*8)   xml   = azimuthal orbital ang-mom quantum number
C  INPUT : (R*8)   xms   = azimuthal spin ang-mom quantum number
C  INPUT : (R*8)   xn1   = principal quantum number (ket state)
C  INPUT : (R*8)   xl1   = orbital angular momentum quantum number
C  INPUT : (R*8)   xml1  = azimuthal orbital ang-mom quantum number
C  INPUT : (R*8)   xms1  = azimuthal spin ang-mom quantum number
C
C  OUTPUT: (R*8)   er    = real part of rel-spin matrix elem (Ryd)
C  OUTPUT: (R*8)   ei    = imag part of rel-spin matrix elem (Ryd)
C
C
C ROUTINES:
C          none
C
C AUTHOR:  Hugh Summers,  University of Strathclyde
C          JA7.08
C          Tel. 0141-548-4196
C
C DATE  :  24/01/06
C
C
C VERSION  : 1.1
C DATE     : 24-01-2006
C MODIFIED : Hugh Summers
C             - First version.
C
C-----------------------------------------------------------------------
      REAL*8              EI,          ER,          XL,          XL1
      REAL*8              XML,         XML1,        XMS,         XMS1
      REAL*8              XN,          XN1
```

4.26 c6ajtb: Subroutine c6ajtb from library adas3xx

```
      SUBROUTINE C6AJTB( MXJSHL , IZ1 , NU , LU , NL , LL , AA )
C
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: C6AJTB *********************
C
C  PURPOSE:  CALCULATES HYDRONIC LJ RESOLVED A-VALUES.
C
C            THE SUBROUTINE CHECKS TO SEE IF A-VALUE IS POSSIBLE AND
C            DIPOLE ALLOWED AND RETURNS ZEROES IF NOT.
C
C  CALLING PROGRAM: ADAS306
C
C  INPUT : (I*4)  MXJSHL   = MAXIMUM NUMBER OF J SUB-SHELLS.
C  INPUT : (I*4)  IZ1      = ION CHARGE.
C  INPUT:  (I*4)  NU       = UPPER PRINCIPAL QUANTUM NUMBER.
C  INPUT:  (I*4)  LU       = ORBITAL QUANTUM NUMBER FOR NU.
C  INPUT:  (I*4)  NL       = LOWER PRINCIPAL QUANTUM NUMBER.
C  INPUT:  (I*4)  LL       = ORBITAL QUANTUM NUMBER FOR NL.
C
C  OUTPUT: (R*8)  AA()     = LJ RESOLVED A-VALUE.
C                            DIMENSION: TRANSITION INDEX WHERE:
C                                       1 GIVES LU+0.5 --> LL+0.5
C                                       2 GIVES LU+0.5 --> LL-0.5
C                                       3 GIVES LU-0.5 --> LL+0.5
C                                       4 GIVES LU-0.5 --> LL-0.5
C
C          (I*4)  I        = LOOP INDEX.
C
C          (R*8)  A        = L RESOLVED A VALUE.
C          (R*8)  XLU      = REAL VALUE = LU.
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          R8ATAB     ADAS      RETURNS L RESOLVED HYDRONIC A-VALUE.
C
C AUTHOR:  JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 5183
C
C DATE:    08/11/93
C
C UNIX-IDL PORT:
C
C AUTHOR:  WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C
C DATE:    22ND MAY 1996
C
C VERSION: 1.1                          DATE: 22-05-96
C MODIFIED: WILLIAM OSBORN
C               - FIRST VERSION. IBM VERSION NOT CHANGED
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IZ1,         LL,          LU,          MXJSHL
      INTEGER             NL,          NU
```

```
      REAL*8              AA(2*MXJSHL)
```

4.27 c6chrg: Subroutine c6chrg from library adas3xx

```
      SUBROUTINE C6CHRG( SYMBD , IZD   , SYMBR , IZR   , IDZ0  ,
     &                   IRZ0  , IRZ1  , IRZ2  , NGRND , NBOT
     &                 )
C
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: C6CHRG *********************
C
C  PURPOSE:  SETS UP NUCLEAR CHARGE OF DONOR AND NULEAR, INITIAL AND
C            FINAL CHARGES OF RECEIVER. CHECKS VALIDITY OF RECEIVER
C            CHARGES. ALSO SETS GROUND STATE N LEVEL AND LOWEST N LEVEL
C            FOR TABULAR OUTPUTS.
C
C  CALLING PROGRAM: ADAS306
C
C  INPUT : (C*2)  SYMBD   = DONOR ELEMENT SYMBOL.
C  INPUT : (I*4)  IZD     = DONOR ION CHARGE.
C  INPUT : (C*2)  SYMBR   = RECEIVER ELEMENT SYMBOL.
C  INPUT : (I*4)  IZR     = RECEIVER ION CHARGE.
C
C  OUTPUT: (I*4)  IDZ0    = DONOR NUCLEAR CHARGE.
C  OUTPUT: (I*4)  IRZ0    = RECEIVER NUCLEAR CHARGE.
C  OUTPUT: (I*4)  IRZ1    = RECEIVER ION INITIAL CHARGE.
C  OUTPUT: (I*4)  IRZ2    = RECEIVER ION FINAL CHARGE.
C  OUTPUT: (I*4)  NGRND   = PRINCIPAL QUANTUM NUMBER OF GROUND STATE.
C  OUTPUT: (I*4)  NBOT    = MINIMUM PRINCIPAL QUANTUM NUMBER.
C
C ROUTINES:
C           ROUTINE     SOURCE      BRIEF DESCRIPTION
C           ------------------------------------------------------------
C           I4UNIT      ADAS        RETURNS UNIT NO. FOR OUTPUT MESSAGES
C           CXCHRG      ADAS        RETURNS DONOR NUCLEAR CHARGE AND
C                                   RECEIVER NULEAR, INITIAL AND FINAL
C                                   CHARGES.
C
C AUTHOR:  JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 5183
C
C DATE:    11/11/93
C
C UNIX-IDL PORT:
C
C AUTHOR:  WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C
C DATE:    22ND MAY 1996
C
C VERSION: 1.1                          DATE: 22-05-96
C MODIFIED: WILLIAM OSBORN
C               - FIRST VERSION. NO CHANGES FROM IBM VERSION.
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      CHARACTER*2         SYMBD,       SYMBR
      INTEGER             IDZ0,        IRZ0,        IRZ1,        IRZ2
      INTEGER             IZD,         IZR,         NBOT,        NGRND
```

4.28 c6emis: Subroutine c6emis from library adas3xx

```
      SUBROUTINE C6EMIS( MXNSHL , MXJSHL , MXOBSL , MXPRSL ,
     &                   IZ0    , IZ1    , NGRND  , NTOT   ,
     &                   NBOT   , DENSZ  , DENS   , NOLINE ,
     &                   NU     , NL     , EMISA  , NPLINE ,
     &                   NPU    , NPL    , QTHEOR , FTHEOR ,
     &                   QTHIN  , TBQMEP , TBQMEM , TBQMIP ,
     &                   TBQMIM , TBFMP  , TBFM   , TBFMM  ,
     &                   NUMIN  , NUMAX  , EM     , QEX    ,
     &                   TOTPOP , TOTEMI , AVRGWL , QEFF   ,
     &                   TBLPOP , TBLEMI , TBLWLN
     &                 )
C
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: C6EMIS *********************
C
C  PURPOSE: PREDICTS THE J-RESOLVED EMMISIVITY FOR REQUESTED
C           TRANSITIONS.
C
C  CALLING PROGRAM: ADAS306
C
C  INPUT : (I*4)  MXNSHL    = MAXIMUM NUMBER OF N SHELLS.
C  INPUT : (I*4)  MXJSHL    = MAXIMUM NUMBER OF J SUB-SHELLS.
C  INPUT : (I*4)  MXOBSL    = MAXIMUM NUMBER OF OBSERVED SPECTRUM
C                             LINES.
C  INPUT : (I*4)  MXPRSL    = MAXIMUM NUMBER OF SPECTRUM LINES TO
C                             PREDICT.
C  INPUT : (I*4)  IZ0       = NUCLEAR CHARGE.
C  INPUT : (I*4)  IZ1       = ION CHARGE.
C  INPUT : (I*4)  NGRND     = PRINCIPAL QUANTUM NUMBER OF GROUND STATE.
C  INPUT : (I*4)  NTOT      = PRINCIPAL QUANTUM NUMBER OF HIGHEST BOUND
C                             STATE.
C  INPUT : (I*4)  NBOT      = MINIMUM PRINCIPAL QUANTUM NUMBER.
C  INPUT : (R*8)  DENSZ     = PLASMA ION DENSITY.
C                             UNITS: CM-3
C  INPUT : (R*8)  DENS      = ELECTRON DENSITY.
C                             UNITS: CM-3
C  INPUT : (I*4)  NOLINE    = NUMBER OF OBSERVED SPECTRUM LINES.
C  INPUT : (I*4)  NU()      = LIST OF UPPER PRINCIPAL QUANTUM NUMBERS
C                             OF OBSERVED SPECTRUM LINES.
C                             DIMENSION: SPECTRUM LINE INDEX.
C  INPUT : (I*4)  NL()      = LIST OF LOWER PRINCIPAL QUANTUM NUMBERS
C                             OF OBSERVED SPECTRUM LINES.
C                             DIMENSION: SPECTRUM LINE INDEX.
C  INPUT : (R*8)  EMISA()   = LIST OF EMISSIVITIES OF OBSERVED SPECTRUM
C                             LINES.
C                             DIMENSION: SPECTRUM LINE INDEX.
C  INPUT : (I*4)  NPLINE    = NUMBER OF SPECTRUM LINES TO PREDICT.
C  INPUT : (I*4)  NPU()     = LIST OF UPPER PRINCIPAL QUANTUM NUMBERS
C                             OF SPECTRUM LINES TO PREDICT.
C                             DIMENSION: PREDICTED LINE INDEX.
C  INPUT : (I*4)  NPL()     = LIST OF LOWER PRINCIPAL QUANTUM NUMBERS
C                             OF SPECTRUM LINES TO PREDICT.
C                             DIMENSION: PREDICTED LINE INDEX.
C  INPUT : (R*8)  QTHEOR()  = MEAN CHARGE EXCHANGE, EXCITATION RATE OR
C                             RECOMBINATION RATE COEFFICIENTS FOR
C                             N-LEVELS AVERAGED OVER BEAM FRACTIONS.
C                             UNITS: CM3 SEC-1
```

```
C                                DIMENSION: N SHELL INDEX.
C  INPUT : (R*8)  FTHEOR(,) = FRACTION OF N-LEVEL MEAN CHARGE EXCHANGE,
C                              EXCITATION RATE OR RECOMBINATION RATE
C                              COEFFICIENTS IN NL-LEVEL.
C                              1ST DIMENSION: J SHELL INDEX WHERE:
C                                             1 GIVES J=L+0.5
C                                             2 GIVES J=L-0.5
C                              2ND DIMENSION: REFERENCED BY I4IDFL(N,L).
C  INPUT : (R*8)  QTHIN()    = IONISATION RATE COEFFICIENT.
C                              UNITS: CM3 SEC-1
C                              DIMENSION: N SHELL INDEX.
C  INPUT : (R*8)  TBQMEP(,) = ELECTRON COLLISIONAL RATE COEFFT. FOR
C                              NLJ->NL+1J'.
C                              1ST DIMENSION: J->J' TRANSITION INDEX.
C                              2ND DIMENSION: REFERENCED BY I4IDFL(N,L).
C  INPUT : (R*8)  TBQMEM(,) = ELECTRON COLLISIONAL RATE COEFFT. FOR
C                              NLJ->NL-1J'.
C                              1ST DIMENSION: J->J' TRANSITION INDEX.
C                              2ND DIMENSION: REFERENCED BY I4IDFL(N,L).
C  INPUT : (R*8)  TBQMIP(,) = POSITIVE ION COLLISIONAL RATE COEFFT. FOR
C                              NLJ->NL+1J'.
C                              1ST DIMENSION: J->J' TRANSITION INDEX.
C                             2 ND DIMENSION: REFERENCED BY I4IDFL(N,L).
C  INPUT : (R*8)  TBQMIM(,) = POSITIVE ION COLLISIONAL RATE COEFFT. FOR
C                              NLJ->NL-1J'.
C                              1ST DIMENSION: J->J' TRANSITION INDEX.
C                              2ND DIMENSION: REFERENCED BY I4IDFL(N,L).
C  INPUT : (R*8)  TBFMP(,)  = B-FIELD DEPENDENT MIXING RATE COEFFT. FOR
C                              NLJ->NL+1J'.
C                              1ST DIMENSION: J->J' TRANSITION INDEX.
C                              2ND DIMENSION: REFERENCED BY I4IDFL(N,L).
C  INPUT : (R*8)  TBFM(,)   = B-FIELD DEPENDENT MIXING RATE COEFFT. FOR
C                              NLJ->NLJ'.
C                              1ST DIMENSION: J->J' TRANSITION INDEX.
C                              2ND DIMENSION: REFERENCED BY I4IDFL(N,L).
C  INPUT : (R*8)  TBFMM(,)  = B-FIELD DEPENDENT MIXING RATE COEFFT. FOR
C                              NLJ->NL-1J'.
C                              1ST DIMENSION: J->J' TRANSITION INDEX.
C                              2ND DIMENSION: REFERENCED BY I4IDFL(N,L).
C
C  OUTPUT: (I*4)  NUMIN     = MINIMUM UPPER PRINCIPAL QUANTUM NUMBER
C                              FOR OBSERVED SPECTRUM LINES.
C  OUTPUT: (I*4)  NUMAX     = MAXIMUM UPPER PRINCIPAL QUANTUM NUMBER
C                              FOR OBSERVED SPECTRUM LINES.
C  OUTPUT: (R*8)  EM        = EMMISSION MEASURE.
C  OUTPUT: (R*8)  QEX()     =
C                              DIMENSION: MXNSHL.
C  OUTPUT: (R*8)  TOTPOP()  = TOTAL COLLISION POP. FOR PREDICTED
C                              SPECTRUM LINE.
C                              UNITS: CM-2
C                              DIMENSION: PREDICTED LINE INDEX.
C  OUTPUT: (R*8)  TOTEMI()  = TOTAL COLLISION EMISSIVITIES FOR PREDICTED
C                              SPECTRUM LINE.
C                              UNITS: PH CM-2 SEC-1
C                              DIMENSION: PREDICTED LINE INDEX.
C  OUTPUT: (R*8)  AVRGWL()  = AVERAGE AIR WAVELENGTH FOR PREDICTED
C                              SPECTRUM LINE.
C                              UNITS: A
C                              DIMENSION: PREDICTED LINE INDEX.
C  OUTPUT: (R*8)  QEFF()    = EFF. RATE COEFFICIENT FOR PREDICTED
C                              SPECTRUM LINE.
```

```
C                               UNITS:
C                               DIMENSION: PREDICTED LINE INDEX.
C  OUTPUT: (R*8)  TBLPOP(,,)= TABLE OF COLLISION POP. FOR PREDICTED
C                               SPECTRUM LINE.
C                               UNITS: CM-2
C                               1ST DIMENSION: J->J' TRANSITION INDEX.
C                               2ND DIMENSION: REFERENCED BY I4IDLI().
C  OUTPUT: (R*8)  TBLEMI(,,)= TABLE OF COLLISION EMISSIVITIES FOR
C                               PREDICTED SPECTRUM LINE.
C                               UNITS: PH CM-2 SEC-1
C                               1ST DIMENSION: J->J' TRANSITION INDEX.
C                               2ND DIMENSION: REFERENCED BY I4IDLI().
C                               3RD DIMENSION: PREDICTED LINE INDEX.
C  OUTPUT: (R*8)  TBLWLN(,,)= TABLE OF WAVELENGTHS FOR PREDICTED
C                               SPECTRUM LINE.
C                               UNITS: A
C                               1ST DIMENSION: J->J' TRANSITION INDEX.
C                               2ND DIMENSION: REFERENCED BY I4IDLI().
C                               3RD DIMENSION: PREDICTED LINE INDEX.
C
C  PARAM : (I*4)  MXN       = MXNSHL.
C  PARAM : (I*4)  MXJ       = MXJSHL.
C  PARAM : (I*4)  MXOB      = MXOBSL.
C
C          (I*4)  NREP      =
C          (I*4)  IC        = LOOP INDEX.
C
C          (I*4)  ICREP()   =
C                               DIMENSION: MXOB.
C
C          (R*8)  WHIGH(,)  =
C                               1ST DIMENSION: J SHELL INDEX.
C                               2ND DIMENSION: REFERENCED BY L+1.
C          (R*8)  WLOW(,,)  =
C                               1ST DIMENSION: J SHELL INDEX.
C                               2ND DIMENSION: REFERENCED BY I4IDFL(N,L).
C                               3RD DIMENSION: REFERENCED BY L+1.
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4UNIT     ADAS      RETURNS UNIT NO. FOR OUTPUT OF MESSAGES.
C          C6WFIL     ADAS
C          C6EMQX
C          C6PRSL     ADAS      PREDICTS REQUESTED SPECTRUM LINES.
C
C NOTES:
C       1) THE J->J' TRANSITION INDEX IS AS FOLLOWS:
C              1 : J=L+0.5 -> J'=L'+0.5
C              2 : J=L+0.5 -> J'=L'-0.5
C              3 : J=L-0.5 -> J'=L'+0.5
C              4 : J=L-0.5 -> J'=L'-0.5
C
C AUTHOR:  JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 5183
C
C DATE:    10/11/93
C
C UNIX-IDL PORT:
C
```

```
C AUTHOR:  WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C
C DATE:    22ND MAY 1996
C
C VERSION: 1.1                          DATE: 22-05-96
C MODIFIED: WILLIAM OSBORN
C               - FIRST VERSION. IBM VERSION NOT CHANGED
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IZ0,          IZ1,          MXJSHL,       MXNSHL
      INTEGER             MXOBSL,       MXPRSL,       NBOT,         NGRND
      INTEGER             NL(MXOBSL),   NOLINE,       NPL(MXPRSL),  NPLINE
      INTEGER             NPU(MXPRSL),  NTOT,         NU(MXOBSL),   NUMAX
      INTEGER             NUMIN
      REAL*8              AVRGWL(MXPRSL),           DENS,         DENSZ
      REAL*8              EM,           EMISA(MXOBSL)
      REAL*8              FTHEOR(MXJSHL,(MXNSHL*(MXNSHL+1))/2)
      REAL*8              QEFF(MXPRSL),             QEX(MXNSHL)
      REAL*8              QTHEOR(MXNSHL),           QTHIN(MXNSHL)
      REAL*8              TBFM(2*MXJSHL,(MXNSHL*(MXNSHL+1))/2)
      REAL*8              TBFMM(2*MXJSHL,(MXNSHL*(MXNSHL+1))/2)
      REAL*8              TBFMP(2*MXJSHL,(MXNSHL*(MXNSHL+1))/2)
      REAL*8              TBLEMI(2*MXJSHL,2*MXNSHL-3,MXPRSL)
      REAL*8              TBLPOP(2*MXJSHL,2*MXNSHL-3,MXPRSL)
      REAL*8              TBLWLN(2*MXJSHL,2*MXNSHL-3,MXPRSL)
      REAL*8              TBQMEM(2*MXJSHL,(MXNSHL*(MXNSHL+1))/2)
      REAL*8              TBQMEP(2*MXJSHL,(MXNSHL*(MXNSHL+1))/2)
      REAL*8              TBQMIM(2*MXJSHL,(MXNSHL*(MXNSHL+1))/2)
      REAL*8              TBQMIP(2*MXJSHL,(MXNSHL*(MXNSHL+1))/2)
      REAL*8              TOTEMI(MXPRSL),           TOTPOP(MXPRSL)
```

4.29 c6pmin: Subroutine c6pmin from library adas3xx

```
      SUBROUTINE C6PMIN( MXNSHL , MXJSHL , N , VD , VDS , VDI , RHS )
C
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: C6PMIN *********************
C
C  PURPOSE:  CALCULATES THE SOLUTION OF A TRIDIAGONAL PARTITIONED MATRIX
C            ORGANISED SET OF SIMULTANEOUS EQUATIONS.
C
C            THE PARTITIONS ARE 2X2 IN THE PRESENT IMPLEMENTATION. A
C            VARIANT OF OF THE DOUBLE PASS ALGORITHM (IN A PARTITIONED
C            MATRIX SENSE) IS USED WITH RECURRENCE IN TWO DIRECTIONS TO
C            THE CENTRE AND BACK. THIS IS ANALOGOUS TO THE NAG ROUTINE
C            F04EAF FOR AN ORDINARY TRIDAGONAL MATRIX THE INDEXING OF
C            THE DIAG, SUPRADIAG AND INFRADIAG ELEMENTS FOLLOWS THAT OF
C            THE NAG ROUTINE F04EAF.
C
C  CALLING PROGRAM: C6WSOL
C
C  INPUT : (I*4)  MXNSHL   = MAXIMUM NUMBER OF N SHELLS.
C  INPUT : (I*4)  MXJSHL   = MAXIMUM NUMBER OF J SUB-SHELLS.
C  INPUT : (I*4)  N        = NUMBER OF PARTIONS ALONG THE DIAGONAL.
C  INPUT : (R*8)  VD(,)    = DIAGONAL PARTITION.
C                            1ST DIMENSION: 2 * MXJSHL
C                            2ND DIMENSION: N SHELL INDEX.
C  INPUT : (R*8)  VDS(,)   = SUPRADIAGONAL PARTITION.
C                            1ST DIMENSION: 2 * MXJSHL
C                            2ND DIMENSION: N SHELL INDEX.
C  INPUT : (R*8)  VDI(,)   = INFRADIAGONAL PARTITION.
C                            1ST DIMENSION: 2 * MXJSHL
C                            2ND DIMENSION: N SHELL INDEX.
C
C  I/O   : (R*8)  RHS(,)   = INPUT: RIGHT HAND SIDE OF VECTOR PARTITION.
C                            OUTPUT: SOLUTION OF VECTOR PARTITION.
C                            1ST DIMENSION: 2 * MXJSHL
C                            2ND DIMENSION: N SHELL INDEX.
C
C
C  PARAM : (I*4)  MXJ      = MXJSHL.
C
C          (I*4)  I        = LOOP INDEX.
C          (I*4)  K        =
C
C          (R*8)  UNIT()   =
C                            DIMENSION: 4
C          (R*8)  W1()     =
C                            DIMENSION: 4
C          (R*8)  W2()     =
C                            DIMENSION: 4
C          (R*8)  VW1()    =
C                            DIMENSION: 2
C          (R*8)  VW2()    =
C                            DIMENSION: 2
C          (R*8)  TEMP()   = TEMPORARY STORE.
C                            DIMENSION: 4
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
```

```
C          ------------------------------------------------------------
C          I4UNIT     ADAS      RETURNS UNIT NO. FOR OUTPUT OF MESSAGES.
C
C NOTES:
C        1) THE 2X2 PARTITIONS ARE STORED AS LINEAR VECTORS BY COLUMN
C           IN THE 1ST DIMENSIONS OF VD(,), VDS(,), VDI(,).
C
C AUTHOR:  JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 5183
C
C DATE:    11/11/93
C
C UNIX-IDL PORT:
C
C AUTHOR:  WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C
C DATE:    22ND MAY 1996
C
C VERSION: 1.1                          DATE: 22-05-96
C MODIFIED: WILLIAM OSBORN
C               - FIRST VERSION. IBM VERSION NOT CHANGED
C
C VERSION: 1.2                          DATE: 29-05-96
C MODIFIED: WILLIAM OSBORN
C               - REMOVED UNUSED VARIABLES
C
C VERSION: 1.3                          DATE: 29-05-96
C MODIFIED: WILLIAM OSBORN
C               - S.C.C.S. MISTAKE
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             MXJSHL,      MXNSHL,      N
      REAL*8              RHS(MXJSHL,MXNSHL),       VD(2*MXJSHL,MXNSHL)
      REAL*8              VDI(2*MXJSHL,MXNSHL),     VDS(2*MXJSHL,MXNSHL)
```

4.30 c6prsl: Subroutine c6prsl from library adas3xx

```
      SUBROUTINE C6PRSL( MXNSHL , MXJSHL , MXPRSL , IZ0    ,
     &                   IZ1    , NPLINE , NPU    , NPL    ,
     &                   NUMAX  , WHIGH  , WLOW   , EM     ,
     &                   QEX    , TOTPOP , TOTEMI , AVRGWL ,
     &                   QEFF   , TBLPOP , TBLEMI , TBLWLN
     &                 )


C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: C6PRSL *********************
C
C  PURPOSE:  FILLS TABLES FOR REQUESTED PREDICTIONS OF SPECTRUM LINES.
C
C  CALLING PROGRAM: C6EMIS
C
C  INPUT : (I*4)  MXNSHL    = MAXIMUM NUMBER OF N SHELLS.
C  INPUT : (I*4)  MXJSHL    = MAXIMUM NUMBER OF J SUB-SHELLS.
C  INPUT : (I*4)  MXPRSL    = MAXIMUM NUMBER OF SPECTRUM LINES TO
C                             PREDICT.
C  INPUT : (R*8)  IZ0       = NUCLEAR CHARGE.
C  INPUT : (R*8)  IZ1       = ION CHARGE.
C  INPUT : (I*4)  NPLINE    = NUMBER OF SPECTRUM LINES TO PREDICT.
C  INPUT : (I*4)  NPU()     = LIST OF UPPER PRINCIPAL QUANTUM NUMBERS
C                             OF SPECTRUM LINES TO PREDICT.
C                             DIMENSION: PREDICTED LINE INDEX.
C  INPUT : (I*4)  NPL()     = LIST OF LOWER PRINCIPAL QUANTUM NUMBERS
C                             OF SPECTRUM LINES TO PREDICT.
C                             DIMENSION: PREDICTED LINE INDEX.
C  INPUT : (I*4)  NUMAX     = MAXIMUM UPPER PRINCIPAL QUANTUM NUMBER
C                             FOR OBSERVED SPECTRUM LINES.
C  INPUT : (R*8)  WHIGH(,)  =
C                             1ST DIMENSION: J SHELL INDEX.
C                             2ND DIMENSION: REFERENCED BY L+1.
C  INPUT : (R*8)  WLOW(,,)  =
C                             1ST DIMENSION: J SHELL INDEX.
C                             2ND DIMENSION: REFERENCED BY I4IDFL(N,L).
C                             3RD DIMENSION: REFERENCED BY L+1.
C  INPUT : (R*8)  EM        = EMMISION MEASURE.
C  INPUT : (R*8)  QEX()     =
C                             DIMENSION: MXNSHL.
C
C  OUTPUT: (R*8)  TOTPOP()  = TOTAL COLLISION POP. FOR PREDICTED
C                             SPECTRUM LINE.
C                             UNITS: CM-2
C                             DIMENSION: PREDICTED LINE INDEX.
C  OUTPUT: (R*8)  TOTEMI()  = TOTAL COLLISION EMISSIVITIES FOR
C                             PREDICTED
C                             SPECTRUM LINE.
C                             UNITS: PH CM-2 SEC-1
C                             DIMENSION: PREDICTED LINE INDEX.
C  OUTPUT: (R*8)  AVRGWL()  = AVERAGE AIR WAVELENGTH FOR PREDICTED
C                             SPECTRUM LINE.
C                             UNITS: A
C                             DIMENSION: PREDICTED LINE INDEX.
C  OUTPUT: (R*8)  QEFF()    = EFF. RATE COEFFICIENT FOR PREDICTED
C                             SPECTRUM LINE.
C                             UNITS:
C                             DIMENSION: PREDICTED LINE INDEX.
```

```
C  OUTPUT: (R*8)  TBLPOP(,,)= TABLE OF COLLISION POP. FOR PREDICTED
C                             SPECTRUM LINE.
C                             UNITS: CM-2
C                             1ST DIMENSION: J->J' TRANSITION INDEX.
C                             2ND DIMENSION: REFERENCED BY I4IDLI().
C  OUTPUT: (R*8)  TBLEMI(,,)= TABLE OF COLLISION EMISSIVITIES FOR
C                             PREDICTED SPECTRUM LINE.
C                             UNITS: PH CM-2 SEC-1
C                             1ST DIMENSION: J->J' TRANSITION INDEX.
C                             2ND DIMENSION: REFERENCED BY I4IDLI().
C                             3RD DIMENSION: PREDICTED LINE INDEX.
C  OUTPUT: (R*8)  TBLWLN(,,)= TABLE OF WAVELENGTHS FOR PREDICTED
C                             SPECTRUM LINE.
C                             UNITS: A
C                             1ST DIMENSION: J->J' TRANSITION INDEX.
C                             2ND DIMENSION: REFERENCED BY I4IDLI().
C                             3RD DIMENSION: PREDICTED LINE INDEX.
C
C  PARAM : (I*4)  C1        = PRECISION AIR WAVELENGTH PARAM.
C  PARAM : (I*4)  C2        = PRECISION AIR WAVELENGTH PARAM.
C  PARAM : (I*4)  C3        = PRECISION AIR WAVELENGTH PARAM.
C  PARAM : (I*4)  C4        = PRECISION AIR WAVELENGTH PARAM.
C  PARAM : (I*4)  C5        = PRECISION AIR WAVELENGTH PARAM.
C  PARAM : (I*4)  RZ        = PRECISION AIR WAVELENGTH PARAM.
C
C          (I*4)  IN        = LOOP INDEX FOR SPECTRUM LINES.
C          (I*4)  N         = PRINCIPAL QUANTUM NUMBER OF INITIAL
C                             STATE.
C          (I*4)  L         = LOOP INDEX FOR ORBITAL QUANTUM NUMBER OF
C          (I*4)  J         = LOOP INDEX FOR J QUANTUM NUMBER OF
C                             INITIAL STATE.
C          (I*4)  N1        = PRINCIPAL QUANTUM NUMBER OF FINAL STATE.
C          (I*4)  L1        = LOOP INDEX FOR ORBITAL QUANTUM NUMBER OF
C                             FINAL STATE.
C          (I*4)  J1        = LOOP INDEX FOR J QUANTUM NUMBER OF
C                             FINAL STATE.
C          (I*4)  NP        = LOOP INDEX FOR PRINCIPAL QUANTUM NUMBER.
C          (I*4)  IDL       = TABLE INDEX.
C          (I*4)  ID        = TABLE INDEX.
C
C          (R*8)  Z1        = REAL VALUE = IZ1.
C          (R*8)  ZEFF1     = EFFECTIVE ION CHARGE.
C          (R*8)  ZEFF2     = EFFECTIVE ION CHARGE.
C          (R*8)  ZEFF3     = EFFECTIVE ION CHARGE.
C          (R*8)  SUM1      = SUM OF COL. POP. FOR PREDICTED LINE.
C                             UNITS: CM-2
C          (R*8)  SUM2      = SUM OF COL. EMIS. FOR PREDICTED LINE.
C                             UNITS: PH CM-2 SEC-1
C          (R*8)  SUM3      = SUM OF WAVELENGTHS FOR PREDICTED LINE.
C                             UNITS: A
C          (R*8)  EU0       = BINDING ENERGY
C                             UNITS: RYD
C          (R*8)  T1        = COL. POP. FOR PREDICTED SPECTRUM LINE.
C                             UNITS: CM-2
C          (R*8)  T2        = COL. EMIS. FOR PREDICTED SPECTRUM LINE.
C                             UNITS: PH CM-2 SEC-1
C          (R*8)  DELTA     =
C          (R*8)  SIG2      =
C          (R*8)  RF        =
C          (R*8)  WAVAIR    = WAVELENGTH FOR PREDICTED SPECTRUM LINE.
C                             UNITS: A
```

```
C
C          (R*8)  EL()       = BINDING ENERGY.
C                               UNITS: RYD
C                               DIMENSION: J SHELL INDEX WHERE:
C                                          1 => L+0.5
C                                          2 => L-0.5
C          (R*8)  EUJ()      = INITIAL STATE J RESOLVED ENERGY.
C                               DIMENSION: J SHELL INDEX WHERE:
C                                          1 => L+0.5
C                                          2 => L-0.5
C          (R*8)  AA()       = LJ RESOLVED A-VALUE.
C                               DIMENSION: TRANSITION INDEX WHERE:
C                                          1 => L+0.5 --> L'+0.5
C                                          2 => L+0.5 --> L'-0.5
C                                          3 => L-0.5 --> L'+0.5
C                                          4 => L-0.5 --> L'-0.5
C
C          (R*8)  ELJ(,)     = FINAL STATE J RESOLVED ENERGY.
C                               1ST DIMENSION: L SHELL INDEX WHERE:
C                                              1 => L'=L+1
C                                              2 => L'=L-1
C                               2ND DIMENSION: J SHELL INDEX WHERE:
C                                              1 => J'=L'+0.5
C                                              2 => J'=L'-0.5
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4UNIT     ADAS      RETURNS UNIT NO. FOR OUTPUT OF MESSAGES.
C          I4IDFL     ADAS      RETURNS UNIQUE INDEX GIVEN QUANTUM
C                               NUMBERS N AND L.
C          I4IDLI     ADAS      RETURNS INDEX FOR PREDICTED SPECTRUM
C                               LINE TABLES.
C          R8ZETA     ADAS
C          C6AJTB     ADAS      RETURNS LJ RESOLVED A-VALUES.
C
C NOTES:
C       1) THE J->J' TRANSITION INDEX IS AS FOLLOWS:
C              1 : J=L+0.5 -> J'=L'+0.5
C              2 : J=L+0.5 -> J'=L'-0.5
C              3 : J=L-0.5 -> J'=L'+0.5
C              4 : J=L-0.5 -> J'=L'-0.5
C
C AUTHOR:  JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 5183
C
C DATE:    09/11/93
C
C UNIX-IDL PORT:
C
C AUTHOR:  WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C
C DATE:    22ND MAY 1996
C
C VERSION: 1.1                          DATE: 22-05-96
C MODIFIED: WILLIAM OSBORN
C               - FIRST VERSION. IBM VERSION NOT CHANGED
C
C
C VERSION: 1.2                          DATE: 08-04-99
```

```
C MODIFIED: Martin O'Mullane
C           - Corrected the vacuum to air conversion. Parameter C4 of the
C             expression for the refractive index of air should be 146.0
C             and not 176.0. (see Astrophysical Quantities section 54)
C           - Introduced R8CONST to return fundamental atomic constants.
C             The mass dependent rydberg constant is returned if set
C             by XXRAMS in the main program. As a consequence RZ has
C             been removed from the parameter statement.
C
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IZ0,          IZ1,          MXJSHL,      MXNSHL
      INTEGER             MXPRSL,       NPL(MXPRSL),  NPLINE
      INTEGER             NPU(MXPRSL),  NUMAX
      REAL*8              AVRGWL(MXPRSL),             EM
      REAL*8              QEFF(MXPRSL),               QEX(MXNSHL)
      REAL*8              TBLEMI(2*MXJSHL,2*MXNSHL-3,MXPRSL)
      REAL*8              TBLPOP(2*MXJSHL,2*MXNSHL-3,MXPRSL)
      REAL*8              TBLWLN(2*MXJSHL,2*MXNSHL-3,MXPRSL)
      REAL*8              TOTEMI(MXPRSL),             TOTPOP(MXPRSL)
      REAL*8              WHIGH(MXJSHL,(MXNSHL*(MXNSHL+1))/2)
      REAL*8              WLOW(MXJSHL,(MXNSHL*(MXNSHL+1))/2,MXNSHL)
```

4.31 c6qeik: Subroutine c6qeik from library adas3xx

```
      SUBROUTINE C6QEIK( MXNSHL , MXJSHL , MXBEAM , IZ1    ,
     &                   IDONOR , NBOT   , NTOP   , NBEAM  ,
     &                   BMENA  , BMFRA  , QTHCH  , FTHCHJ
     &                 )
C
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: C6QEIK *********************
C
C  PURPOSE:  USES THE EIKONIAL APPROXIMATION TO CALCULATE THE
C            THEORETICAL CHARGE EXCHANGE RATE COEFFICIENTS TO N SHELLS
C            AND THE NLJ FRACTIONS FROM NEUTRAL HYDROGEN OR HELIUM IN
C            GROUND OR EXCITED STATE TO A BARE NUCLEUS TARGET.
C
C  CALLING PROGRAM: ADAS306
C
C  INPUT : (I*4)  MXNSHL    = MAXIMUM NO. OF N SHELLS.
C  INPUT : (I*4)  MXJSHL    = MAXIMUM NO. OF J SUB-SHELLS.
C  INPUT : (I*4)  MXBEAM    = MAXIMUM NO. OF BEAM ENERGIES.
C  INPUT : (R*8)  IZ1       = CHARGE OF TARGET ION.
C  INPUT : (I*4)  IDONOR    = DONOR STATE FOR EIKONAL MODEL.
C                             1 = H(1S)    DONOR
C                             2 = H(2S)    DONOR
C                             3 = H(2P)    DONOR
C                             4 = H(1S2)   DONOR
C                             5 = HE(1S2S) DONOR
C  INPUT : (I*4)  NBOT      = MINIMUM PRINCIPAL QUANTUM NUMBER.
C  INPUT : (I*4)  NTOP      = MAXIMUM PRINCIPAL QUANTUM NUMBER.
C  INPUT : (I*4)  NBEAM     = NUMBER OF BEAM ENERGIES.
C  INPUT : (R*8)  BMENA()   = BEAM ENERGY COMPONENTS.
C                             UNITS: EV/AMU
C                             DIMENSION: COMPONENT INDEX.
C  INPUT : (R*8)  BMFRA()   = BEAM COMPONENT FRACTIONS.
C                             DIMENSION: COMPONENT INDEX.
C
C  OUTPUT: (R*8)  QTHCH()   = MEAN RATE COEFFICIENTS FOR N-LEVELS
C                             AVERAGED OVER BEAM FRACTIONS.
C                             UNITS: CM3 SEC-1
C                             DIMENSION: REFERENCED BY N QUANTUM NUMBER.
C  OUTPUT: (R*8)  FTHCHJ(,) = MEAN RATE COEFFICIENTS FOR NLJ-LEVELS
C                             AVERAGED OVER BEAM FRACTIONS. EXPRESSED AS
C                             FRACTIONS OF CORRESPONDING N-LEVELS.
C                             1ST DIMENSION: J SUB-SHELL
C                                            1 => J=L+0.5
C                                            2 => J=L-0.5
C                             2ND DIMENSION: REFERENCED BY I4IDFL(N,L)
C
C  PARAM : (I*4)  MXN       = 'MXNSHL'.
C
C          (I*4)  N         = N-SHELL INDEX.
C          (I*4)  L         = L-SHELL INDEX.
C          (I*4)  J         = J-SHELL INDEX.
C          (I*4)  IDL       = L-RESOLVED INDEX.
C
C          (R*8)  XL        = REAL VALUE = L.
C          (R*8)  WL        =
C
C          (R*8)  FTHCH()   = MEAN RATE COEFFICIENTS FOR NL-LEVELS
```

```
C                             AVERAGED OVER BEAM FRACTIONS. EXPRESSED AS
C                             FRACTIONS OF CORRESPONDING N-LEVELS.
C                             DIMENSION: REFERENCED BY I4IDFL(N,L)
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4UNIT     ADAS      RETURNS UNIT NO. FOR OUTPUT OF MESSAGES.
C          I4IDFL     ADAS      RETURNS UNIQUE INDEX GIVEN QUANTUM
C                               NUMBERS N AND L.
C          CXQEIK     ADAS      CALCULATES N-LEVEL AND NL-LEVEL CHARGE
C                               EXCHANGE RATE COEFFICIENTS USING EIKONAL
C                               APPROX. NL RATES ARE GIVEN AS FRACTION
C                               OF CORRESPONDING N RATE.
C
C AUTHOR:  JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 5183
C
C DATE:    22/10/93
C
C UNIX-IDL PORT:
C
C AUTHOR:  WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C
C DATE:    22ND MAY 1996
C
C VERSION: 1.1                          DATE: 22-05-96
C MODIFIED: WILLIAM OSBORN
C               - FIRST VERSION. IBM VERSION NOT CHANGED
C
C VERSION: 1.2                          DATE: 17-05-07
C MODIFIED: Allan Whiteford
C               - Fixed typo in comments
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IDONOR,      IZ1,         MXBEAM,      MXJSHL
      INTEGER             MXNSHL,      NBEAM,       NBOT,        NTOP
      REAL*8              BMENA(MXBEAM),            BMFRA(MXBEAM)
      REAL*8              FTHCHJ(MXJSHL,(MXNSHL*(MXNSHL+1))/2)
      REAL*8              QTHCH(MXNSHL)
```

4.32 c6qxch: Subroutine c6qxch from library adas3xx

```
      SUBROUTINE C6QXCH ( MXNENG , MXNSHL , MXJSHL , MXBEAM ,
     &                    NBEAM  , BMENA  , BMFRA  , NBOT   ,
     &                    NTOP   , NMINF  , NMAXF  , NENRGY ,
     &                    ENRGYA , ALPHAA , XSECNA , FRACLA ,
     &                    QTHCH  , FTHCHJ
     &                  )
C
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: C6QXCH *********************
C
C  PURPOSE:  USES THE INPUT DATASET TO CALCULATE THE CHARGE EXCHANGE
C            RATE COEFFICIENTS FOR BOTH N-LEVELS AND NLJ-LEVELS AVERAGED
C            OVER THE BEAM FRACTIONS.
C
C            NLJ-LEVEL RATES ARE EXPRESSED AS A FRACTION OF
C            CORRESPONDING N-LEVEL.
C
C  CALLING PROGRAM: ADAS306
C
C  INPUT : (I*4)  MXNENG    = MAXIMUM NO. OF ENERGIES.
C  INPUT : (I*4)  MXNSHL    = MAXIMUM NO. OF N SHELLS.
C  INPUT : (I*4)  MXJSHL    = MAXIMUM NO. OF J SUB-SHELLS.
C  INPUT : (I*4)  MXBEAM    = MAXIMUM NO. OF BEAM ENERGIES.
C  INPUT : (I*4)  NBEAM     = NUMBER OF BEAM ENERGIES.
C  INPUT : (R*8)  BMENA()   = BEAM ENERGY COMPONENTS.
C                             UNITS: EV/AMU
C                             DIMENSION: COMPONENT INDEX.
C  INPUT : (R*8)  BMFRA()   = BEAM COMPONENT FRACTIONS.
C                             DIMENSION: COMPONENT INDEX.
C  INPUT : (I*4)  NBOT      = MINIMUM PRINCIPAL QUANTUM NUMBER.
C  INPUT : (I*4)  NTOP      = MAXIMUM PRINCIPAL QUANTUM NUMBER.
C  INPUT : (I*4)  NMINF     = MINIMUM PRINCIPAL QUANTUM NUMBER OF INPUT
C                             DATASET.
C  INPUT : (I*4)  NMAXF     = MAXIMUM PRINCIPAL QUANTUM NUMBER OF INPUT
C                             DATASET.
C  INPUT : (I*4)  NENRGY    = NUMBER OF ENERGIES IN DATASET.
C  INPUT : (R*8)  ENRGYA()  = COLLISION ENERGIES.
C                             UNITS: EV/AMU
C                             DIMENSION: ENERGY INDEX
C  INPUT : (R*8)  ALPHAA()  = EXTRAPOLATION PARAMETER ALPHA.
C                             DIMENSION: ENERGY INDEX
C  INPUT : (R*8)  XSECNA(,) = N-RESOLVED CHARGE EXCHANGE CROSS-SECTIONS.
C                             UNITS: CM2
C                             1ST DIMENSION: ENERGY INDEX
C                             2ND DIMENSION: N-SHELL
C  INPUT : (R*8)  FRACLA(,) = L-RESOLVED CHARGE EXCHANGE CROSS-SECTIONS
C                             EXPRESSED AS FRACTION OF CORRESPONDING
C                             N-RESOLVED CROSS-SECTION.
C                             1ST DIMENSION: ENERGY INDEX
C                             2ND DIMENSION: REFERENCED BY I4IDFL(N,L)
C
C  OUTPUT: (R*8)  QTHCH()   = MEAN RATE COEFFICIENTS FOR N-LEVELS
C                             AVERAGED OVER BEAM FRACTIONS.
C                             UNITS: CM3 SEC-1
C                             DIMENSION: REFERENCED BY N QUANTUM NUMBER.
C  OUTPUT: (R*8)  FTHCHJ(,) = MEAN RATE COEFFICIENTS FOR NLJ-LEVELS
C                             AVERAGED OVER BEAM FRACTIONS. EXPRESSED AS
```

```
C                             FRACTIONS OF CORRESPONDING N-LEVELS.
C                             1ST DIMENSION: J SUB-SHELL
C                                            1 => J=L+0.5
C                                            2 => J=L-0.5
C                             2ND DIMENSION: REFERENCED BY I4IDFL(N,L)
C
C  PARAM : (I*4)  MXN       = 'MXNSHL'.
C
C          (I*4)  N         = N-SHELL INDEX.
C          (I*4)  L         = L-SHELL INDEX.
C          (I*4)  J         = J-SHELL INDEX.
C          (I*4)  IDL       = L-RESOLVED INDEX.
C
C          (R*8)  XL        = REAL VALUE = L.
C          (R*8)  WL        =
C
C          (R*8)  FTHCH()   = MEAN RATE COEFFICIENTS FOR NL-LEVELS
C                             AVERAGED OVER BEAM FRACTIONS. EXPRESSED AS
C                             FRACTIONS OF CORRESPONDING N-LEVELS.
C                             DIMENSION: REFERENCED BY I4IDFL(N,L)
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4UNIT     ADAS      RETURNS UNIT NO. FOR OUTPUT OF MESSAGES.
C          I4IDFL     ADAS      RETURNS UNIQUE INDEX GIVEN QUANTUM
C                               NUMBERS N AND L.
C          CXQXCH     ADAS      CALCULATES N-LEVEL AND NL-LEVEL CHARGE
C                               EXCHANGE RATE COEFFICIENTS USING INPUT
C                               DATASET. NL RATES ARE GIVEN AS FRACTION
C                               OF CORRESPONDING N RATE.
C
C AUTHOR:  JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 5183
C
C DATE:    22/10/93
C
C UNIX-IDL PORT:
C
C AUTHOR:  WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C
C DATE:    22ND MAY 1996
C
C VERSION: 1.1                          DATE: 22-05-96
C MODIFIED: WILLIAM OSBORN
C               - FIRST VERSION. IBM VERSION NOT CHANGED
C
C VERSION: 1.2                          DATE: 17-05-07
C MODIFIED: Allan Whiteford
C               - Updated comments as part of subroutine documentation
C                 procedure.
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             MXBEAM,      MXJSHL,      MXNENG,      MXNSHL
      INTEGER             NBEAM,       NBOT,        NENRGY,      NMAXF
      INTEGER             NMINF,       NTOP
      REAL*8              ALPHAA(MXNENG),           BMENA(MXBEAM)
      REAL*8              BMFRA(MXBEAM),            ENRGYA(MXNENG)
      REAL*8              FRACLA(MXNENG,(MXNSHL*(MXNSHL+1))/2)
```

```
REAL*8              FTHCHJ(MXJSHL,(MXNSHL*(MXNSHL+1))/2)
REAL*8              QTHCH(MXNSHL)
REAL*8              XSECNA(MXNENG,MXNSHL)
```

4.33 c6tbex: Subroutine c6tbex from library adas3xx

```
      SUBROUTINE C6TBEX( MXNSHL , MXJSHL , IZ1    , NBOT   ,
     &                   NTOP   , NGRND  , TEV    , QTHEX  ,
     &                   FTHEXJ
     &                 )
C
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: C6TBEX *********************
C
C  PURPOSE:  SETS UP A TABLE OF ELECTRON IMPACT EXCITATION RATE
C            COEFFICIENTS FOR A HYDROGENIC ION FROM THE GROUND STATE
C            TO EXCITED NL LEVELS.
C
C  CALLING PROGRAM: ADAS306.
C
C  INPUT : (I*4)  MXNSHL    = MAXIMUM NO. OF N SHELLS.
C  INPUT : (I*4)  MXJSHL    = MAXIMUM NO. J SUB-SHELLS.
C  INPUT : (I*4)  IZ1       = ION CHARGE.
C  INPUT : (I*4)  NBOT      = MINIMUM PRINCIPAL QUANTUM NUMBER.
C  INPUT : (I*4)  NTOP      = MAXIMUM PRINCIPAL QUANTUM NUMBER.
C  INPUT : (I*4)  NGRND     =
C  INPUT : (R*8)  TEV       = ELECTRON TEMPERATURE.
C                             UNITS: EV
C
C  OUTPUT: (R*8)  QTHEX()   = TABLE OF N-LEVEL EXCITATION RATE
C                             COEFFICIENTS.
C                             UNITS:
C                             DIMENSION: N-SHELL
C  OUTPUT: (R*8)  FTHEXJ(,) = TABLE OF NLJ-LEVEL EXCITATION RATE
C                             COEFFICIENTS EXPRESSED AS FRACTION OF
C                             CORRESPONDING N-LEVEL RATE.
C                             1ST DIMENSION: J SUB-SHELL
C                                            1 => J=L+0.5
C                                            2 => J=L-0.5
C                             2ND DIMENSION: REFERENCED BY I4IDFL(N,L)
C
C  PARAM : (I*4)  MXN       = 'MXNSHL'.
C
C          (I*4)  N         = N-SHELL INDEX.
C          (I*4)  L         = L-SHELL INDEX.
C          (I*4)  J         = J-SHELL INDEX.
C          (I*4)  IDL       = L-RESOLVED INDEX.
C
C          (R*8)  XL        = REAL VALUE = L.
C          (R*8)  WL        =
C
C          (R*8)  TBQEX()   = TABLE OF NL-LEVEL EXCITATION RATE
C                             COEFFICIENTS.
C                             UNITS:
C                             DIMENSION: REFERENCED BY I4IDFL(N,L).
C          (R*8)  FTHEX()   = TABLE OF NL-LEVEL EXCITATION RATE
C                             COEFFICIENTS EXPRESSED AS FRACTION OF
C                             CORRESPONDING N-LEVEL RATE.
C                             DIMENSION: REFERENCED BY I4IDFL(N,L).
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
```

```
C          I4UNIT     ADAS      RETURNS UNIT NO. FOR OUTPUT OF MESSAGES.
C          I4IDFL     ADAS      RETURNS UNIQUE INDEX GIVEN QUANTUM
C                               NUMBERS N AND L.
C          CXTBEX     ADAS      CALCULATES N-LEVEL AND NL-LEVEL
C                               EXCITATION RATE COEFFICIENTS. NL RATES
C                               ARE GIVEN AS FRACTION OF CORRESPONDING
C                               N RATE.
C
C AUTHOR:   JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C           K1/0/81
C           JET EXT. 5183
C
C DATE:     22/10/93
C
C UNIX-IDL PORT:
C
C AUTHOR:  WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C
C DATE:    22ND MAY 1996
C
C VERSION: 1.1                          DATE: 22-05-96
C MODIFIED: WILLIAM OSBORN
C               - FIRST VERSION. IBM VERSION NOT CHANGED
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IZ1,         MXJSHL,      MXNSHL,      NBOT
      INTEGER             NGRND,       NTOP
      REAL*8              FTHEXJ(MXJSHL,(MXNSHL*(MXNSHL+1))/2)
      REAL*8              QTHEX(MXNSHL),            TEV
```

4.34 c6tbfm: Subroutine c6tbfm from library adas3xx

```
      SUBROUTINE C6TBFM( MXNSHL , MXJSHL , IZ0    , IZ1    ,
     &                   AMSSNO , NBOT   , NTOP   , BMAG   ,
     &                   TIEV   , TBLF   , TBFMP  , TBFM   ,
     &                   TBFMM
     &                 )
C
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: C6TBFM *********************
C
C  PURPOSE:  FILLS TABLES OF  MAGNETIC FIELD DEPENDENT MIXING RATE
C            COEFFICIENTS BETWEEN NEARLY DEGENERATE LEVELS FOR
C            HYDROGEN-LIKE, LITHIUM-LIKE AND SODIUM-LIKE IONS.
C
C            RATES ARE CALCULATED FOR THE SEPARATE NLJ->NL+1J',
C            NLJ->NLJ' AND NLJ->NL-1J' TRANSITIONS.
C
C  CALLING PROGRAM: ADAS306
C
C  INPUT : (I*4)  MXNSHL  = MAXIMUM NUMBER OF N SHELLS.
C  INPUT : (I*4)  MXJSHL  = MAXIMUM NUMBER OF J SUB-SHELLS.
C  INPUT : (I*4)  IZ0     = TARGET NUCLEAR CHARGE.
C  INPUT : (I*4)  IZ1     = ION CHARGE.
C  INPUT : (R*8)  AMSSNO  = ATOMIC MASS NO.
C  INPUT : (I*4)  NBOT    = MINIMUM PRINCIPAL QUANTUM NUMBER.
C  INPUT : (I*4)  NTOP    = MAXIMUM PRINCIPAL QUANTUM NUMBER.
C  INPUT : (R*8)  BMAG    = MAGNETIC INDUCTION.
C                           UNITS: TESLA
C  INPUT : (R*8)  TIEV    = TEMPERATURE (ION DISTRIBUTION).
C                           UNITS: EV
C  INPUT : (R*8)  TBLF()  = TABLE OF RADIATIVE LIFETIMES.
C                           UNITS: SECS
C                           DIMENSION: REFERENCED BY FUNC I4IDFL(N,L).
C
C  OUTPUT: (R*8)  TBFMP(,)= RATE COEFFT. FOR NLJ->NL+1J'.
C                           1ST DIMENSION: J->J' TRANSITION INDEX.
C                           2ND DIMENSION: REFERENCED BY I4IDFL(N,L).
C  OUTPUT: (R*8)  TBFM(,) = RATE COEFFT. FOR NLJ->NL+1J'.
C                           1ST DIMENSION: J->J' TRANSITION INDEX.
C                           2ND DIMENSION: REFERENCED BY I4IDFL(N,L).
C  OUTPUT: (R*8)  TBFMM(,)= RATE COEFFT. FOR NLJ->NLJ' FOR STATE I.
C                           1ST DIMENSION: J->J' TRANSITION INDEX.
C                           2ND DIMENSION: REFERENCED BY I4IDFL(N,L).
C
C  PARAM : (I*4)  MXJ     = 'MXJSHL'.
C
C          (I*4)  NI      = VALENCE ELECTRON PRINCIPAL QUANTUM NUMBER
C                           IN STATE I.
C          (I*4)  NJ      = VALENCE ELECTRON PRINCIPAL QUANTUM NUMBER
C                           IN STATE J.
C          (I*4)  LI      = VALENCE ELECTRON ORBITAL QUANTUM NUMBER IN
C                           STATE I.
C          (I*4)  LJ      = VALENCE ELECTRON ORBITAL QUANTUM NUMBER IN
C                           STATE J.
C          (I*4)  IDLI    = TABLE INDEX.
C          (I*4)  IDLJ    = TABLE INDEX.
C          (I*4)  I       = LOOP INDEX.
C          (I*4)  J       = LOOP INDEX.
```

```
C
C          (R*8)  FMP()    = RATE COEFFT. FOR NLJ->NL+1J'.
C                             DIMENSION: J->J' TRANSITION INDEX.
C          (R*8)  FMM()    = RATE COEFFT. FOR NLJ->NL+1J'.
C                             DIMENSION: J->J' TRANSITION INDEX.
C          (R*8)  FMI()    = RATE COEFFT. FOR NLJ->NLJ' FOR STATE I.
C                             DIMENSION: J->J' TRANSITION INDEX.
C          (R*8)  FMJ()    = RATE COEFFT. FOR NLJ->NLJ' FOR STATE J.
C                             DIMENSION: J->J' TRANSITION INDEX.
C
C NOTES:
C       1) THE J->J' TRANSITION INDEX IS AS FOLLOWS:
C              1 : J=L+0.5 -> J'=L'+0.5
C              2 : J=L+0.5 -> J'=L'-0.5
C              3 : J=L-0.5 -> J'=L'+0.5
C              4 : J=L-0.5 -> J'=L'-0.5
C
C       2) BEFORE CALLING C6TBQM THE LIFETIME TABLE MUST BE FILLED
C          WITH A CALL TO C6TBLF.
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4UNIT     ADAS      RETURNS UNIT NO. FOR OUTPUT OF MESSAGES.
C          I4IDFL     ADAS      RETURNS UNIQUE INDEX GIVEN QUANTUM
C                               NUMBERS N AND L.
C          CXMRDG     ADAS      CALCULATES MIXING RATE COEFFICIENTS
C                               BETWEEN NEARLY DEGENERATE LEVELS OF
C                               H-, LI- OR NA-LIKE IONS.
C
C AUTHOR:  JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 5183
C
C DATE:    04/11/93
C
C UNIX-IDL PORT:
C
C AUTHOR:  WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C
C DATE:    22ND MAY 1996
C
C VERSION: 1.1                          DATE: 22-05-96
C MODIFIED: WILLIAM OSBORN
C               - FIRST VERSION. IBM VERSION NOT CHANGED
C
C VERSION: 1.2                          DATE: 17-05-07
C MODIFIED: Allan Whiteford
C               - Updated comments as part of subroutine documentation
C                 procedure.
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IZ0,         IZ1,         MXJSHL,      MXNSHL
      INTEGER             NBOT,        NTOP
      REAL*8              AMSSNO,      BMAG
      REAL*8              TBFM(2*MXJSHL,(MXNSHL*(MXNSHL+1))/2)
      REAL*8              TBFMM(2*MXJSHL,(MXNSHL*(MXNSHL+1))/2)
      REAL*8              TBFMP(2*MXJSHL,(MXNSHL*(MXNSHL+1))/2)
      REAL*8              TBLF((MXNSHL*(MXNSHL+1))/2),           TIEV
```

4.35 c6tbin: Subroutine c6tbin from library adas3xx

```
      SUBROUTINE C6TBIN( MXNSHL , IZ1 , NBOT , NTOP , TEV , QTHIN )
C
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: C6TBIN *********************
C
C  PURPOSE:  SETS UP AN ELECTRON IMPACT IONISATION RATE COEFFICIENT
C            TABLE FOR N-LEVELS BASED ON THE ECIP APPROXIMATION. ENERGY
C            LEVELS ARE ASSUMED HYDROGENIC IN THE EFFECTIVE ION CHARGE.
C
C  CALLING PROGRAM: ADAS306
C
C  INPUT : (I*4)  MXNSHL   = MAXIMUM VALUE OF PRINCIPAL QUANTUM NUMBER.
C  INPUT : (I*4)  IZ1      = ION CHARGE.
C  INPUT : (I*4)  NBOT     = MINIMUM PRINCIPAL QUANTUM NUMBER.
C  INPUT : (I*4)  NTOP     = MAXIMUM PRINCIPAL QUANTUM NUMBER.
C  INPUT : (R*8)  TEV      = ELECTRON TEMPERATURE.
C                            UNITS: EV
C
C  OUTPUT: (R*8)  QTHIN()  = IONISATION RATE COEFFICIENT.
C                            UNITS: CM3 SEC-1
C                            DIMENSION: N SHELL INDEX.
C
C  PARAM : (R*8)  P1       =
C
C          (I*4)  N        = LOOP INDEX.
C          (I*4)  IZC      = IZ1-1.
C
C          (R*8)  ZETA     = EFFECTIVE NUMBER OF EQUIVALENT ELECTRONS FOR
C                            SHELL.
C          (R*8)  TE       = ELECTRON TEMPERATURE.
C                            UNITS: K
C          (R*8)  XI       =
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          R8ECIP     ADAS
C
C AUTHOR:  JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 5183
C
C DATE:    03/11/93
C
C UNIX-IDL PORT:
C
C AUTHOR:  WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C
C DATE:    22ND MAY 1996
C
C VERSION: 1.1                          DATE: 22-05-96
C MODIFIED: WILLIAM OSBORN
C               - FIRST VERSION. IBM VERSION NOT CHANGED
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
```

```
INTEGER             IZ1,          MXNSHL,      NBOT,         NTOP
REAL*8              QTHIN(MXNSHL),             TEV
```

4.36 c6tbqm: Subroutine c6tbqm from library adas3xx

```
      SUBROUTINE C6TBQM( MXNSHL , MXJSHL , IZ0    , IZ1    ,
     &                   NBOT   , NTOP   , TEV    , DENS   ,
     &                   ZP     , TPV    , EMP    , TBLF   ,
     &                   TBQMEP , TBQMEM , TBQMIP , TBQMIM
     &                 )
C
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: C6TBQM *********************
C
C  PURPOSE:  SETS UP TABLES OF ELECTRON AND POSITIVE ION COLLISIONAL
C            RATE COEFFICIENTS BETWEEN NEARLY DEGENERATE LEVELS FOR
C            H-, LI-, AND NA-LIKE IONS.
C
C            THE RATES FOR THE SEPARATE TRANSITIONS NLJ->NL+1J' AND
C            NLJ->NL-1J' ARE OBTAINED.
C
C  CALLING PROGRAM: ADAS306
C
C  INPUT : (I*4)  MXNSHL   = MAXIMUM VALUE OF PRINCIPAL QUANTUM NUMBER.
C  INPUT : (I*4)  MXJSHL   = MAXIMUM NUMBER OF J SUB-SHELLS.
C  INPUT : (I*4)  IZ0      = NUCLEAR CHARGE OF TARGET ION.
C  INPUT : (I*4)  IZ1      = ION CHARGE.
C  INPUT : (I*4)  NBOT     = MINIMUM PRINCIPAL QUANTUM NUMBER.
C  INPUT : (I*4)  NTOP     = MAXIMUM PRINCIPAL QUANTUM NUMBER.
C  INPUT : (R*8)  TEV      = ELECTRON TEMPERATURE.
C                            UNITS: EV
C  INPUT : (R*8)  DENS     = ELECTRON DENSITY.
C                            UNITS: CM-3
C  INPUT : (R*8)  ZP       = CHARGE OF COLLIDING POSITIVE ION.
C  INPUT : (R*8)  TPV      = TEMPERATURE (COLLIDING POSITIVE ION
C                            DISTRIBUTION).
C                            UNITS: EV
C  INPUT : (R*8)  EMP      = REDUCED MASS FOR COLLIDING POSITIVE ION.
C                            UNITS: ELECTRON MASSES
C  INPUT : (R*8)  ZP       = CHARGE OF COLLIDING POSITIVE ION.
C  INPUT : (R*8)  TPV      = POSITIVE ION TEMPERATURE.
C                            UNITS: EV
C  INPUT : (R*8)  EMP      = REDUCED MASS FOR COLLIDING POSITIVE ION.
C                            UNITS: ELECTRON MASSES
C  INPUT : (R*8)  TBLF()   = TABLE OF RADIATIVE LIFETIMES.
C                            UNITS: SECS
C                            DIMENSION: REFERENCED BY FUNC I4IDFL(N,L).
C
C  OUTPUT: (R*8)  TBQMEP(,)= ELECTRON RATE COEFFT. FOR NLJ->NL+1J'.
C                            1ST DIMENSION: J->J' TRANSITION INDEX.
C                            2ND DIMENSION: REFERENCED BY I4IDFL(N,L).
C  OUTPUT: (R*8)  TBQMEM(,)= ELECTRON RATE COEFFT. FOR NLJ->NL-1J'.
C                            1ST DIMENSION: J->J' TRANSITION INDEX.
C                            2ND DIMENSION: REFERENCED BY I4IDFL(N,L).
C  OUTPUT: (R*8)  TBQMIP(,)= POSITIVE ION RATE COEFFT. FOR NLJ->NL+1J'.
C                            1ST DIMENSION: J->J' TRANSITION INDEX.
C                            2ND DIMENSION: REFERENCED BY I4IDFL(N,L).
C  OUTPUT: (R*8)  TBQMIM(,)= POSITIVE ION RATE COEFFT. FOR NLJ->NL-1J'.
C                            1ST DIMENSION: J->J' TRANSITION INDEX.
C                            2ND DIMENSION: REFERENCED BY I4IDFL(N,L).
C
C  PARAM : (I*4)  MXJ     = 'MXJSHL'.
```

```
C
C          (I*4)  NI      = VALENCE ELECTRON PRINCIPAL QUANTUM NUMBER
C                           IN STATE I.
C          (I*4)  NJ      = VALENCE ELECTRON PRINCIPAL QUANTUM NUMBER
C                           IN STATE J.
C          (I*4)  LI      = VALENCE ELECTRON ORBITAL QUANTUM NUMBER IN
C                           STATE I.
C          (I*4)  LJ      = VALENCE ELECTRON ORBITAL QUANTUM NUMBER IN
C                           STATE J.
C          (I*4)  IDLI    = TABLE INDEX.
C          (I*4)  IDLJ    = TABLE INDEX.
C          (I*4)  I       = LOOP INDEX.
C          (I*4)  J       = LOOP INDEX.
C
C          (R*8)  GAE     = GAMA RATE PARAMETER FOR ELECTRON COLLISIONS.
C          (R*8)  GAP     = GAMA RATE PARAMETER FOR POSITIVE ION
C                           COLLISIONS.
C
C          (R*8)  QEP()   = ELECTRON RATE COEFFT. FOR NLJ->NL+1J'
C                           DIMENSION: J->J' TRANSITION INDEX.
C          (R*8)  QEM()   = ELECTRON RATE COEFFT. FOR NLJ->NL-1J'
C                           DIMENSION: J->J' TRANSITION INDEX.
C          (R*8)  QIP()   = POSITIVE ION RATE COEFFT. FOR NLJ->NL+1J'
C                           DIMENSION: J->J' TRANSITION INDEX.
C          (R*8)  QIM()   = POSITIVE ION RATE COEFFT. FOR NLJ->NL-1J'
C                           DIMENSION: J->J' TRANSITION INDEX.
C
C NOTES:
C       1) THE J->J' TRANSITION INDEX IS AS FOLLOWS:
C              1 : J=L+0.5 -> J'=L'+0.5
C              2 : J=L+0.5 -> J'=L'-0.5
C              3 : J=L-0.5 -> J'=L'+0.5
C              4 : J=L-0.5 -> J'=L'-0.5
C
C       2) BEFORE CALLING C6TBQM THE LIFETIME TABLE MUST BE FILLED
C          WITH A CALL TO C6TBLF.
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4UNIT     ADAS      RETURNS UNIT NO. FOR OUTPUT OF MESSAGES.
C          I4IDFL     ADAS      RETURNS UNIQUE INDEX GIVEN QUANTUM
C                               NUMBERS N AND L.
C          CXCRDG     ADAS      CALCULATES COLLISIONAL RATE COEFFICIENTS
C                               BETWEEN NEARLY DEGENERATE LEVELS OF
C                               H-, LI- OR NA-LIKE IONS.
C
C AUTHOR:  JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 5183
C
C DATE:    02/11/93
C
C UNIX-IDL PORT:
C
C AUTHOR:  WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C
C DATE:    22ND MAY 1996
C
C VERSION: 1.1                          DATE: 22-05-96
C MODIFIED: WILLIAM OSBORN
```

```
C               - FIRST VERSION. IBM VERSION NOT CHANGED
C
C VERSION: 1.2                          DATE: 17-05-07
C MODIFIED: Allan Whiteford
C               - Updated comments as part of subroutine documentation
C                 procedure.
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IZ0,          IZ1,          MXJSHL,      MXNSHL
      INTEGER             NBOT,         NTOP
      REAL*8              DENS,         EMP
      REAL*8              TBLF((MXNSHL*(MXNSHL+1))/2)
      REAL*8              TBQMEM(2*MXJSHL,(MXNSHL*(MXNSHL+1))/2)
      REAL*8              TBQMEP(2*MXJSHL,(MXNSHL*(MXNSHL+1))/2)
      REAL*8              TBQMIM(2*MXJSHL,(MXNSHL*(MXNSHL+1))/2)
      REAL*8              TBQMIP(2*MXJSHL,(MXNSHL*(MXNSHL+1))/2)
      REAL*8              TEV,          TPV,          ZP
```

4.37 c6tbrc: Subroutine c6tbrc from library adas3xx

```
      SUBROUTINE C6TBRC( MXNSHL , MXJSHL , IZ1    , NBOT   ,
     &                   NTOP   , TEV    , QTHRC  , FTHRCJ
     &                 )
C
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: C6TBRC *********************
C
C  PURPOSE:  SETS UP A TABLE OF RADIATIVE RECOMBINATION RATE
C            COEFFICIENTS FOR A BARE NUCLEUS, HELIUM-LIKE OR NEON-LIKE
C            ION TO EXCITED NLJ LEVELS.
C
C  CALLING PROGRAM: ADAS306
C
C  INPUT : (I*4)  MXNSHL   = MAXIMUM VALUE OF PRINCIPAL QUANTUM NUMBER.
C  INPUT : (I*4)  MXJSHL   = MAXIMUM NUMBER OF J SUB-SHELLS.
C  INPUT : (I*4)  IZ1      = ION CHARGE.
C  INPUT : (I*4)  NBOT     = MINIMUM PRINCIPAL QUANTUM NUMBER.
C  INPUT : (I*4)  NTOP     = MAXIMUM PRINCIPAL QUANTUM NUMBER.
C  INPUT : (R*8)  TEV      = ELECTRON TEMPERATURE.
C                            UNITS: EV
C
C  OUTPUT: (R*8)  QTHRC()  = RECOMBINATION RATE COEFFICIENT TO LEVEL N.
C                            UNITS: CM3 SEC-1
C                            DIMENSION: N-SHELL
C  OUTPUT: (R*8)  FTHRCJ(,)= FRACTION OF RECOMBINATION RATE OF LEVEL N
C                            TO STATE NLJ.
C                            1ST DIMENSION: J-SHELL INDEX WHERE
C                                           1 GIVES J=L+0.5
C                                           2 GIVES J=L-0.5
C                            2ND DIMENSION: REFERENCED BY I4IDFL().
C
C  PARAM : (R*8)  P1       = BOLTZMANN CONSTANT.
C                            UNITS: EV K-1
C  PARAM : (R*8)  P2       =
C
C          (I*4)  N        = PRINCIPAL QUANTUM NUMBER OF BOUND ELECTRON.
C          (I*4)  L        = ORBITAL QUANTUM NUMBER OF BOUND ELECTRON.
C          (I*4)  L1       = ORBITAL QUANTUM NUMBER OF FREE ELECTRON.
C          (I*4)  LP       = TOTAL ORBITAL ANGULAR MOMENTUM QUANTUM
C                            NUMBER OF PARENT STATE.
C          (I*4)  ISP      = 2*SP+1 WHERE SP IS TOTAL SPIN OF PAREN     T
C                            STATE.
C          (I*4)  LT       = TOTAL ORBITAL ANGULAR MOMENTUM QUANTUM
C                            NUMBER OF BOUND SYSTEM.
C          (I*4)  LT1      = TOTAL ORBITAL ANGULAR MOMENTUM QUANTUM
C                            NUMBER OF FREE SYSTEM.
C          (I*4)  IS       = 2*S+1 WHERE S IS TOTAL SPIN OF SYSTEM.
C          (I*4)  IRES     = LEVEL OF RESOLUTION.
C                          = 1 :
C                          = 2 : ABOVE LT1 SUM.
C                          = 3 : ABOVE LT SUM.
C                          = 4 : ABOVE S SUM.
C                          = 5 : UNRESOLVED GBF.
C          (I*4)  I        = LOOP INDEX.
C          (I*4)  J        = LOOP INDEX.
C          (I*4)  IDL      = TABLE INDEX.
C
```

```
C          (R*8)  Z1       = REAL VALUE = IZ1.
C          (R*8)  TE       = ELECTRON TEMPERATURE.
C                            UNITS: K
C          (R*8)  V        = EFFECTIVE PRINCIPAL QUANTUM NUMBER OF BOUND
C                            ELECTRON.
C          (R*8)  FACT     =
C          (R*8)  SUM      =
C          (R*8)  XL       = REAL VALUE = L.
C          (R*8)  WL       =
C          (R*8)  T        =
C          (R*8)  PREC1    = RADIATIVE RECOMBINATION INTEGRAL.
C          (R*8)  PION1    = PHOTOIONISATION INTEGRAL.
C          (R*8)  PSTIM1   = STIMULATED RECOMBINATION INTEGRAL.
C          (R*8)  PREC2    = RADIATIVE RECOMBINATION INTEGRAL.
C          (R*8)  PION2    = PHOTOIONISATION INTEGRAL.
C          (R*8)  PSTIM2   = STIMULATED RECOMBINATION INTEGRAL.
C
C  PARAM : (R*8)  P1       = BOLTZMANN CONSTANT.
C                            UNITS: EV K-1
C  PARAM : (R*8)  P2       =
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4IDFL     ADAS      RETURNS UNIQUE INDEX GIVEN QUANTUM
C                               NUMBERS N AND L.
C          CXPHOT     ADAS      CALCULATES PHOTO INTEGRALS USING GIIH
C                               BOUND-FREE GAUNT-FACTORS.
C
C AUTHOR:  JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 5183
C
C DATE:    05/11/93
C
C UNIX-IDL PORT:
C
C AUTHOR:  WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C
C DATE:    22ND MAY 1996
C
C VERSION: 1.1                          DATE: 22-05-96
C MODIFIED: WILLIAM OSBORN
C               - FIRST VERSION. IBM VERSION NOT CHANGED
C
C VERSION: 1.2                          DATE: 29-05-96
C MODIFIED: WILLIAM OSBORN
C               - REMOVED UNUSED VARIABLES
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IZ1,         MXJSHL,      MXNSHL,      NBOT
      INTEGER             NTOP
      REAL*8              FTHRCJ(MXJSHL,(MXNSHL*(MXNSHL+1))/2)
      REAL*8              QTHRC(MXNSHL),            TEV
```

4.38 c7cxee: Subroutine c7cxee from library adas3xx

```
      SUBROUTINE C7CXEE( MXNENG , MXNSHL , NGRND  , NTOT   ,
     &                   NBOT   , NTOP   , IRZ0   , IRZ1   ,
     &                   RAMSNO , TEV    , TIEV   , DENS   ,
     &                   DENSZ  , ZEFF   , BMAG   , BMENG  ,
     &                   ITHEOR , IBSTAT , IEMMS  , NTU    ,
     &                   NTL    , NMINF  , NMAXF  , NENRGY ,
     &                   ENRGYA , ALPHAA , XSECNA , FRACLA ,
     &                   ERATE
     &                 )
C
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: C7CXEE *********************
C
C  PURPOSE:  CALCULATES THE J-RESOLVED EFFECTIVE EMISSIVITY RATE
C            COEFFICIENT FOR THE GIVEN TRANSITION.
C
C            IT IS APPLICABLE TO IMPURITIES IN PLASMA TRAVERSED BY
C            NEUTRAL BEAMS OF H OR HE. THE RECOMBINED TARGET ION MAY BE
C            H, LI OR NA-LIKE.
C
C            THE MODEL INCLUDES CAPTURE, N-N' LEVEL CASCADE, AND MIXING
C            AMONG L,J LEVELS OF SAME N BY COLLISIONS OR MAGNETIC
C            FIELDS.
C
C            ELECTRON IMPACT IONISATION IS INCLUDED TO GIVE COLLISION
C            LIMIT EFFECT.
C
C            AN INTERNAL EIKONAL APPROXIMATION IS USED FOR CAPTURE FROM
C            EXCITED H OR HE STATES, ALTHOUGH NORMALLY THE EXTERNAL DATA
C            SET SHOULD BE USED.
C
C  CALLING PROGRAM: ADAS307
C
C  INPUT : (I*4)  MXNENG    = MAXIMUM NO. OF ENERGIES IN DATA SET.
C  INPUT : (I*4)  MXNSHL    = MAXIMUM NUMBER OF N SHELLS.
C  INPUT : (I*4)  NGRND     = PRINCIPAL QUANTUM NUMBER OF GROUND STATE.
C  INPUT : (I*4)  NTOT      = PRINCIPAL QUANTUM NUMBER OF HIGHEST BOUND
C                             STATE.
C  INPUT : (I*4)  NBOT      = MINIMUM PRINCIPAL QUANTUM NUMBER FOR
C                             RATE TABLES.
C  INPUT : (I*4)  NTOP      = MAXIMUM PRINCIPAL QUANTUM NUMBER FOR
C                             RATE TABLES.
C  INPUT : (I*4)  IRZ0      = RECEIVER NUCLEAR CHARGE.
C  INPUT : (I*4)  IRZ1      = RECEIVER ION INITIAL CHARGE.
C  INPUT : (R*8)  RAMSNO    = RECEIVER ATOMIC MASS.
C  INPUT : (R*8)  TEV       = ELECTRON TEMPERATURE.
C                             UNITS: EV
C  INPUT : (R*8)  TIEV      = ION TEMPERATURE.
C                             UNITS: EV
C  INPUT : (R*8)  DENS      = ELECTRON DENSITY.
C                             UNITS: CM-3
C  INPUT : (R*8)  DENSZ     = PLASMA ION DENSITY.
C                             UNITS: CM-3
C  INPUT : (R*8)  ZEFF      = EFFECTIVE ION CHARGE.
C  INPUT : (R*8)  BMAG      = PLASMA MAGNETIC INDUCTION.
C                             UNITS: TESLA
C  INPUT : (R*8)  BMENG     = BEAM ENERGY.
```

```
C                                UNITS: EV/AMU
C  INPUT : (I*4)  ITHEOR    = CHARGE EXCHANGE MODEL OPTION.
C                                1 => USE INPUT DATA SET.
C                                2 => USE EIKONAL MODEL.
C  INPUT : (I*4)  IBSTAT     = DONOR STATE FOR EIKONAL MODEL.
C                                1 => H(1S)
C                                2 => H(2S)
C                                3 => H(2P)
C                                4 => HE(1S2)
C                                5 => HE(1S2S)
C  INPUT : (I*4)  IEMMS      = EMISSION MEASURE MODEL OPTION.
C                                1 => CHARGE EXCHANGE.
C                                2 => ELECTRON IMPACT EXCITATION.
C                                3 => RADIATIVE RECOMBINATION.
C  INPUT : (I*4)  NTL        = LOWER PRINCIPAL QUANTUM NUMBER OF
C                                TRANSITION.
C  INPUT : (I*4)  NTU        = UPPER PRINCIPAL QUANTUM NUMBER OF
C                                TRANSITION.
C  INPUT : (I*4)  NMINF      = LOWEST N-SHELL FOR WHICH DATA READ.
C  INPUT : (I*4)  NMAXF      = HIGHEST N-SHELL FOR WHICH DATA READ.
C  INPUT : (I*4)  NENRGY     = NUMBER OF ENERGIES READ FROM DATA SET.
C  INPUT : (R*8)  ENRGYA()   = COLLISION ENERGIES READ FROM INPUT DATA
C                                SET.
C                                UNITS: EV/AMU
C                                DIMENSION: ENERGY INDEX
C  INPUT : (R*8)  ALPHAA()   = EXTRAPOLATION PARAMETER ALPHA READ FROM
C                                INPUT DATA SET.
C                                DIMENSION: ENERGY INDEX
C  INPUT : (R*8)  XSECNA(,)  = N-RESOLVED CHARGE EXCHANGE CROSS-SECTIONS
C                                READ FROM INPUT DATA SET.
C                                UNITS: CM2
C                                1ST DIMENSION: ENERGY INDEX
C                                2ND DIMENSION: N-SHELL
C  INPUT : (R*8)  FRACLA(,)  = L-RESOLVED CHARGE EXCHANGE CROSS-SECTIONS.
C                                AFTER CXDATA: ABSOLUTE VALUES (CM2).
C                                AFTER CXFRAC: FRACTION OF N-RESOLVED
C                                              DATA.
C                                1ST DIMENSION: ENERGY INDEX
C                                2ND DIMENSION: INDEXED BY I4IDFL(N,L)
C
C  OUTPUT: (R*8)  ERATE      = EFFECTIVE EMISSIVITY RATE COEFFICIENT FOR
C                                REQUESTED TRANSITION
C                                SPECTRUM LINE.
C                                UNITS: CM3 SEC-1
C
C  PARAM : (I*4)  MXN        = MXNSHL.
C  PARAM : (I*4)  MXJSHL     = MAXIMUM NUMBER OF J SUB-SHELLS.
C  PARAM : (I*4)  MXBEAM     = MAXIMUM NUMBER OF BEAM COMPONENTS.
C  PARAM : (I*4)  MXOBSL     = MAXIMUM NUMBER OF OBSERVED SPECTRUM
C                                LINES.
C  PARAM : (I*4)  MXPRSL     = MAXIMUM NUMBER OF SPECTRUM LINES TO
C                                PREDICT.
C
C  PARAM : (R*8)  EMP        = REDUCED MASS FOR POSITIVE ION.
C                                UNITS: ELECTRON MASSES
C
C          (I*4)  NBEAM      = NUMBER OF BEAM ENERGIES.
C          (I*4)  NOLINE     = NUMBER OF OBSERVED SPECTRUM LINES.
C          (I*4)  NPLINE     = NUMBER OF SPECTRUM LINES TO PREDICT.
C          (I*4)  NUMIN      = MINIMUM UPPER PRINCIPAL QUANTUM NUMBER
C                                FOR OBSERVED SPECTRUM LINES.
```

```
C          (I*4)  NUMAX     = MAXIMUM UPPER PRINCIPAL QUANTUM NUMBER
C                              FOR OBSERVED SPECTRUM LINES.
C
C          (R*8)  EM        = EMMISSION MEASURE.
C                              UNITS: CM-5
C
C          (I*4)  NL()      = LIST OF LOWER PRINCIPAL QUANTUM NUMBERS
C                              OF OBSERVED SPECTRUM LINES.
C                              DIMENSION: SPECTRUM LINE INDEX.
C          (I*4)  NU()      = LIST OF UPPER PRINCIPAL QUANTUM NUMBERS
C                              OF OBSERVED SPECTRUM LINES.
C                              DIMENSION: SPECTRUM LINE INDEX.
C          (I*4)  NPL()     = LIST OF LOWER PRINCIPAL QUANTUM NUMBERS
C                              OF SPECTRUM LINES TO PREDICT.
C                              DIMENSION: SPECTRUM LINE INDEX.
C          (I*4)  NPU()     = LIST OF UPPER PRINCIPAL QUANTUM NUMBERS
C                              OF SPECTRUM LINES TO PREDICT.
C                              DIMENSION: SPECTRUM LINE INDEX.
C
C          (R*8)  BMFRA()   = BEAM COMPONENT FRACTIONS.
C                              DIMENSION: COMPONENT INDEX.
C          (R*8)  BMENA()   = BEAM ENERGY COMPONENTS.
C                              UNITS: EV/AMU
C          (R*8)  EMISA()   = LIST OF EMISSIVITIES OF OBSERVED SPECTRUM
C                              LINES.
C                              UNITS: PH CM-2 SEC-1
C                              DIMENSION: SPECTRUM LINE INDEX.
C          (R*8)  TBLF()    = TABLE OF RADIATIVE LIFETIMES.
C                              UNITS: SECS
C                              DIMENSION: REFERENCED BY I4IDFL(N,L).
C          (R*8)  QTHIN()   = IONISATION RATE COEFFICIENT.
C                              UNITS: CM3 SEC-1
C                              DIMENSION: N SHELL INDEX.
C          (R*8)  QTHEX()   = MEAN EXCITATION RATE COEFFICIENTS FOR
C                              N-LEVELS AVERAGED OVER BEAM FRACTIONS.
C                              UNITS: CM3 SEC-1
C                              DIMENSION: N SHELL INDEX.
C          (R*8)  QTHCH()   = MEAN CHARGE EXCHANGE COEFFICIENTS FOR
C                              N-LEVELS AVERAGED OVER BEAM FRACTIONS.
C                              UNITS: CM3 SEC-1
C                              DIMENSION: N SHELL INDEX.
C          (R*8)  QTHRC()   = MEAN RECOMBINATION RATE COEFFICIENTS FOR
C                              N-LEVELS AVERAGED OVER BEAM FRACTIONS.
C                              UNITS: CM3 SEC-1
C                              DIMENSION: N SHELL INDEX.
C          (R*8)  QEX()     =
C                              DIMENSION: N SHELL INDEX.
C          (R*8)  TOTPOP()  = TOTAL COLLISION POP. FOR PREDICTED
C                              SPECTRUM LINE.
C                              UNITS: CM-2
C                              DIMENSION: PREDICTED LINE INDEX.
C          (R*8)  TOTEMI()  = TOTAL COLLISION EMISSIVITIES FOR PREDICTED
C                              SPECTRUM LINE.
C                              UNITS: PH CM-2 SEC-1
C                              DIMENSION: PREDICTED LINE INDEX.
C          (R*8)  AVRGWL()  = AVERAGE AIR WAVELENGTH FOR PREDICTED
C                              SPECTRUM LINE.
C                              UNITS: A
C                              DIMENSION: PREDICTED LINE INDEX.
C          (R*8)  QEFF()    = EFF. RATE COEFFICIENT FOR PREDICTED
C                              SPECTRUM LINE.
```

```
C                              UNITS: CM3 SEC-1
C                              DIMENSION: PREDICTED LINE INDEX.
C
C          (R*8)  FTHEXJ(,) = FRACTION OF N-LEVEL MEAN EXCITATION RATE
C                              COEFFICIENTS IN NLJ-LEVEL.
C                              1ST DIMENSION: J SHELL INDEX WHERE:
C                                             1 GIVES J=L+0.5
C                                             2 GIVES J=L-0.5
C                              2ND DIMENSION: REFERENCED BY I4IDFL(N,L).
C          (R*8)  FTHCHJ(,) = FRACTION OF N-LEVEL MEAN CHARGE EXCHANGE
C                              COEFFICIENTS IN NLJ-LEVEL.
C                              1ST DIMENSION: J SHELL INDEX WHERE:
C                                             1 GIVES J=L+0.5
C                                             2 GIVES J=L-0.5
C                              2ND DIMENSION: REFERENCED BY I4IDFL(N,L).
C          (R*8)  FTHRCJ(,) = FRACTION OF N-LEVEL MEAN RECOMBINATION
C                              RATE COEFFICIENTS IN NLJ-LEVEL.
C                              1ST DIMENSION: J SHELL INDEX WHERE:
C                                             1 GIVES J=L+0.5
C                                             2 GIVES J=L-0.5
C                              2ND DIMENSION: REFERENCED BY I4IDFL(N,L).
C          (R*8)  TBQMEP(,) = ELECTRON COLLISIONAL RATE COEFFT. FOR
C                              NLJ->NL+1J'.
C                              1ST DIMENSION: J->J' TRANSITION INDEX.
C                              2ND DIMENSION: REFERENCED BY I4IDFL(N,L).
C          (R*8)  TBQMEM(,) = ELECTRON COLLISIONAL RATE COEFFT. FOR
C                              NLJ->NL-1J'.
C                              1ST DIMENSION: J->J' TRANSITION INDEX.
C                              2ND DIMENSION: REFERENCED BY I4IDFL(N,L).
C          (R*8)  TBQMIP(,) = POSITIVE ION COLLISIONAL RATE COEFFT. FOR
C                              NLJ->NL+1J'.
C                              1ST DIMENSION: J->J' TRANSITION INDEX.
C                              2ND DIMENSION: REFERENCED BY I4IDFL(N,L).
C          (R*8)  TBQMIM(,) = POSITIVE ION COLLISIONAL RATE COEFFT. FOR
C                              NLJ->NL-1J'.
C                              1ST DIMENSION: J->J' TRANSITION INDEX.
C                              2ND DIMENSION: REFERENCED BY I4IDFL(N,L).
C          (R*8)  TBFMP(,)  = B-FIELD DEPENDENT MIXING RATE COEFFT. FOR
C                              NLJ->NL+1J'.
C                              1ST DIMENSION: J->J' TRANSITION INDEX.
C                              2ND DIMENSION: REFERENCED BY I4IDFL(N,L).
C          (R*8)  TBFM(,)   = B-FIELD DEPENDENT MIXING RATE COEFFT. FOR
C                              NLJ->NLJ'.
C                              1ST DIMENSION: J->J' TRANSITION INDEX.
C                              2ND DIMENSION: REFERENCED BY I4IDFL(N,L).
C          (R*8)  TBFMM(,)  = B-FIELD DEPENDENT MIXING RATE COEFFT. FOR
C                              NLJ->NL-1J'.
C                              1ST DIMENSION: J->J' TRANSITION INDEX.
C                              2ND DIMENSION: REFERENCED BY I4IDFL(N,L).
C
C          (R*8)  TBLPOP(,,)= TABLE OF COLLISION POP. FOR PREDICTED
C                              SPECTRUM LINE.
C                              UNITS: CM-2
C                              1ST DIMENSION: J->J' TRANSITION INDEX.
C                              2ND DIMENSION: REFERENCED BY I4IDLI().
C                              3RD DIMENSION: PREDICTED LINE INDEX.
C          (R*8)  TBLEMI(,,)= TABLE OF COLLISION EMISSIVITIES FOR
C                              PREDICTED SPECTRUM LINE.
C                              UNITS: PH CM-2 SEC-1
C                              1ST DIMENSION: J->J' TRANSITION INDEX.
C                              2ND DIMENSION: REFERENCED BY I4IDLI().
```

```
C                               3RD DIMENSION: PREDICTED LINE INDEX.
C          (R*8)  TBLWLN(,,)= TABLE OF WAVELENGTHS FOR PREDICTED
C                               SPECTRUM LINE.
C                               UNITS: A
C                               1ST DIMENSION: J->J' TRANSITION INDEX.
C                               2ND DIMENSION: REFERENCED BY I4IDLI().
C                               3RD DIMENSION: PREDICTED LINE INDEX.
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4UNIT     ADAS      RETURNS UNIT NO. FOR OUTPUT OF MESSAGES.
C          CXTBLF     ADAS      FILLS L-RESOLVED RADIATIVE LIFETIME
C                               TABLE.
C          C6TBIN     ADAS      FILLS N-RESOLVED ELECTRON IMPACT
C                               IONISATION RATE TABLE.
C          C6TBEX     ADAS      FILLS N AND J-RESOLVED ELECTRON IMPACT
C                               EXCITATION RATE TABLES.
C          C6QEIK     ADAS      FILLS N AND J-RESOLVED CHARGE EXCHANGE
C                               RATE TABLES USING EIKONAL APPROXIMATION.
C          C6QXCH     ADAS      FILLS N AND J-RESOLVED CHARGE EXCHANGE
C                               RATE TABLES USING INPUT DATA SET.
C          C6TBRC     ADAS      FILLS N AND J-RESOLVED RADIATIVE
C                               RECOMBINATION RATE TABLES.
C          C6TBQM     ADAS      FILLS N AND J-RESOLVED COLLISIONAL RATE
C                               TABLES.
C          C6TBFM     ADAS      FILLS N AND J-RESOLVED B-FIELD
C                               DEPENDENT MIXING RATE TABLES.
C          C7EMIS     ADAS      PREDICTS THE J-RESOLVED EMISSIVITY FOR
C                               REQUESTED TRANSITIONS.
C
C NOTES:
C       1) THE J->J' TRANSITION INDEX IS AS FOLLOWS:
C              1 : J=L+0.5 -> J'=L'+0.5
C              2 : J=L+0.5 -> J'=L'-0.5
C              3 : J=L-0.5 -> J'=L'+0.5
C              4 : J=L-0.5 -> J'=L'-0.5
C
C AUTHOR:  JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/87
C          JET EXT. 5183
C
C DATE:    26/11/93
C
C UNIX-IDL PORT:
C
C AUTHOR:  WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C
C DATE:    24TH MAY 1996
C
C VERSION: 1.1                          DATE: 24-05-96
C MODIFIED: WILLIAM OSBORN
C               - FIRST VERSION
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IBSTAT,      IEMMS,       IRZ0,        IRZ1
      INTEGER             ITHEOR,      MXNENG,      MXNSHL,      NBOT
      INTEGER             NENRGY,      NGRND,       NMAXF,       NMINF
      INTEGER             NTL,         NTOP,        NTOT,        NTU
```

```
REAL*8              ALPHAA(MXNENG),            BMAG,         BMENG
REAL*8              DENS,         DENSZ,       ENRGYA(MXNENG)
REAL*8              ERATE
REAL*8              FRACLA(MXNENG,(MXNSHL*(MXNSHL+1))/2),  RAMSNO
REAL*8              TEV,          TIEV
REAL*8              XSECNA(MXNENG,MXNSHL),     ZEFF
```

4.39 c7emis: Subroutine c7emis from library adas3xx

```
      SUBROUTINE C7EMIS( MXNSHL , MXJSHL , MXOBSL , MXPRSL ,
     &                   IZ0    , IZ1    , NGRND  , NTOT   ,
     &                   NBOT   , DENSZ  , DENS   , NOLINE ,
     &                   NU     , NL     , EMISA  , NPLINE ,
     &                   NPU    , NPL    , QTHEOR , FTHEOR ,
     &                   QTHIN  , TBQMEP , TBQMEM , TBQMIP ,
     &                   TBQMIM , TBFMP  , TBFM   , TBFMM  ,
     &                   NUMIN  , NUMAX  , EM     , QEX    ,
     &                   TOTPOP , TOTEMI , AVRGWL , QEFF   ,
     &                   TBLPOP , TBLEMI , TBLWLN
     &                 )
C
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: C6EMIS *********************
C
C  PURPOSE: PREDICTS THE J-RESOLVED EMMISIVITY FOR REQUESTED
C           TRANSITIONS.
C
C  CALLING PROGRAM: ADAS306
C
C  INPUT : (I*4)  MXNSHL    = MAXIMUM NUMBER OF N SHELLS.
C  INPUT : (I*4)  MXJSHL    = MAXIMUM NUMBER OF J SUB-SHELLS.
C  INPUT : (I*4)  MXOBSL    = MAXIMUM NUMBER OF OBSERVED SPECTRUM
C                             LINES.
C  INPUT : (I*4)  MXPRSL    = MAXIMUM NUMBER OF SPECTRUM LINES TO
C                             PREDICT.
C  INPUT : (I*4)  IZ0       = NUCLEAR CHARGE.
C  INPUT : (I*4)  IZ1       = ION CHARGE.
C  INPUT : (I*4)  NGRND     = PRINCIPAL QUANTUM NUMBER OF GROUND STATE.
C  INPUT : (I*4)  NTOT      = PRINCIPAL QUANTUM NUMBER OF HIGHEST BOUND
C                             STATE.
C  INPUT : (I*4)  NBOT      = MINIMUM PRINCIPAL QUANTUM NUMBER.
C  INPUT : (R*8)  DENSZ     = PLASMA ION DENSITY.
C                             UNITS: CM-3
C  INPUT : (R*8)  DENS      = ELECTRON DENSITY.
C                             UNITS: CM-3
C  INPUT : (I*4)  NOLINE    = NUMBER OF OBSERVED SPECTRUM LINES.
C  INPUT : (I*4)  NU()      = LIST OF UPPER PRINCIPAL QUANTUM NUMBERS
C                             OF OBSERVED SPECTRUM LINES.
C                             DIMENSION: SPECTRUM LINE INDEX.
C  INPUT : (I*4)  NL()      = LIST OF LOWER PRINCIPAL QUANTUM NUMBERS
C                             OF OBSERVED SPECTRUM LINES.
C                             DIMENSION: SPECTRUM LINE INDEX.
C  INPUT : (R*8)  EMISA()   = LIST OF EMISSIVITIES OF OBSERVED SPECTRUM
C                             LINES.
C                             DIMENSION: SPECTRUM LINE INDEX.
C  INPUT : (I*4)  NPLINE    = NUMBER OF SPECTRUM LINES TO PREDICT.
C  INPUT : (I*4)  NPU()     = LIST OF UPPER PRINCIPAL QUANTUM NUMBERS
C                             OF SPECTRUM LINES TO PREDICT.
C                             DIMENSION: PREDICTED LINE INDEX.
C  INPUT : (I*4)  NPL()     = LIST OF LOWER PRINCIPAL QUANTUM NUMBERS
C                             OF SPECTRUM LINES TO PREDICT.
C                             DIMENSION: PREDICTED LINE INDEX.
C  INPUT : (R*8)  QTHEOR()  = MEAN CHARGE EXCHANGE, EXCITATION RATE OR
C                             RECOMBINATION RATE COEFFICIENTS FOR
C                             N-LEVELS AVERAGED OVER BEAM FRACTIONS.
C                             UNITS: CM3 SEC-1
```

```
C                                DIMENSION: N SHELL INDEX.
C  INPUT : (R*8)  FTHEOR(,) = FRACTION OF N-LEVEL MEAN CHARGE EXCHANGE,
C                              EXCITATION RATE OR RECOMBINATION RATE
C                              COEFFICIENTS IN NL-LEVEL.
C                              1ST DIMENSION: J SHELL INDEX WHERE:
C                                             1 GIVES J=L+0.5
C                                             2 GIVES J=L-0.5
C                              2ND DIMENSION: REFERENCED BY I4IDFL(N,L).
C  INPUT : (R*8)  QTHIN()    = IONISATION RATE COEFFICIENT.
C                              UNITS: CM3 SEC-1
C                              DIMENSION: N SHELL INDEX.
C  INPUT : (R*8)  TBQMEP(,) = ELECTRON COLLISIONAL RATE COEFFT. FOR
C                              NLJ->NL+1J'.
C                              1ST DIMENSION: J->J' TRANSITION INDEX.
C                              2ND DIMENSION: REFERENCED BY I4IDFL(N,L).
C  INPUT : (R*8)  TBQMEM(,) = ELECTRON COLLISIONAL RATE COEFFT. FOR
C                              NLJ->NL-1J'.
C                              1ST DIMENSION: J->J' TRANSITION INDEX.
C                              2ND DIMENSION: REFERENCED BY I4IDFL(N,L).
C  INPUT : (R*8)  TBQMIP(,) = POSITIVE ION COLLISIONAL RATE COEFFT. FOR
C                              NLJ->NL+1J'.
C                              1ST DIMENSION: J->J' TRANSITION INDEX.
C                             2 ND DIMENSION: REFERENCED BY I4IDFL(N,L).
C  INPUT : (R*8)  TBQMIM(,) = POSITIVE ION COLLISIONAL RATE COEFFT. FOR
C                              NLJ->NL-1J'.
C                              1ST DIMENSION: J->J' TRANSITION INDEX.
C                              2ND DIMENSION: REFERENCED BY I4IDFL(N,L).
C  INPUT : (R*8)  TBFMP(,)  = B-FIELD DEPENDENT MIXING RATE COEFFT. FOR
C                              NLJ->NL+1J'.
C                              1ST DIMENSION: J->J' TRANSITION INDEX.
C                              2ND DIMENSION: REFERENCED BY I4IDFL(N,L).
C  INPUT : (R*8)  TBFM(,)   = B-FIELD DEPENDENT MIXING RATE COEFFT. FOR
C                              NLJ->NLJ'.
C                              1ST DIMENSION: J->J' TRANSITION INDEX.
C                              2ND DIMENSION: REFERENCED BY I4IDFL(N,L).
C  INPUT : (R*8)  TBFMM(,)  = B-FIELD DEPENDENT MIXING RATE COEFFT. FOR
C                              NLJ->NL-1J'.
C                              1ST DIMENSION: J->J' TRANSITION INDEX.
C                              2ND DIMENSION: REFERENCED BY I4IDFL(N,L).
C
C  OUTPUT: (I*4)  NUMIN     = MINIMUM UPPER PRINCIPAL QUANTUM NUMBER
C                              FOR OBSERVED SPECTRUM LINES.
C  OUTPUT: (I*4)  NUMAX     = MAXIMUM UPPER PRINCIPAL QUANTUM NUMBER
C                              FOR OBSERVED SPECTRUM LINES.
C  OUTPUT: (R*8)  EM        = EMMISSION MEASURE.
C  OUTPUT: (R*8)  QEX()     =
C                              DIMENSION: MXNSHL.
C  OUTPUT: (R*8)  TOTPOP()  = TOTAL COLLISION POP. FOR PREDICTED
C                              SPECTRUM LINE.
C                              UNITS: CM-2
C                              DIMENSION: PREDICTED LINE INDEX.
C  OUTPUT: (R*8)  TOTEMI()  = TOTAL COLLISION EMISSIVITIES FOR PREDICTED
C                              SPECTRUM LINE.
C                              UNITS: PH CM-2 SEC-1
C                              DIMENSION: PREDICTED LINE INDEX.
C  OUTPUT: (R*8)  AVRGWL()  = AVERAGE AIR WAVELENGTH FOR PREDICTED
C                              SPECTRUM LINE.
C                              UNITS: A
C                              DIMENSION: PREDICTED LINE INDEX.
C  OUTPUT: (R*8)  QEFF()    = EFF. RATE COEFFICIENT FOR PREDICTED
C                              SPECTRUM LINE.
```

```
C                              UNITS:
C                              DIMENSION: PREDICTED LINE INDEX.
C  OUTPUT: (R*8)  TBLPOP(,,)= TABLE OF COLLISION POP. FOR PREDICTED
C                              SPECTRUM LINE.
C                              UNITS: CM-2
C                              1ST DIMENSION: J->J' TRANSITION INDEX.
C                              2ND DIMENSION: REFERENCED BY I4IDLI().
C  OUTPUT: (R*8)  TBLEMI(,,)= TABLE OF COLLISION EMISSIVITIES FOR
C                              PREDICTED SPECTRUM LINE.
C                              UNITS: PH CM-2 SEC-1
C                              1ST DIMENSION: J->J' TRANSITION INDEX.
C                              2ND DIMENSION: REFERENCED BY I4IDLI().
C                              3RD DIMENSION: PREDICTED LINE INDEX.
C  OUTPUT: (R*8)  TBLWLN(,,)= TABLE OF WAVELENGTHS FOR PREDICTED
C                              SPECTRUM LINE.
C                              UNITS: A
C                              1ST DIMENSION: J->J' TRANSITION INDEX.
C                              2ND DIMENSION: REFERENCED BY I4IDLI().
C                              3RD DIMENSION: PREDICTED LINE INDEX.
C
C  PARAM : (I*4)  MXN       = MXNSHL.
C  PARAM : (I*4)  MXJ       = MXJSHL.
C  PARAM : (I*4)  MXOB      = MXOBSL.
C
C          (I*4)  NREP      =
C          (I*4)  IC        = LOOP INDEX.
C
C          (I*4)  ICREP()   =
C                              DIMENSION: MXOB.
C
C          (R*8)  WHIGH(,)  =
C                              1ST DIMENSION: J SHELL INDEX.
C                              2ND DIMENSION: REFERENCED BY L+1.
C          (R*8)  WLOW(,,)  =
C                              1ST DIMENSION: J SHELL INDEX.
C                              2ND DIMENSION: REFERENCED BY I4IDFL(N,L).
C                              3RD DIMENSION: REFERENCED BY L+1.
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4UNIT     ADAS      RETURNS UNIT NO. FOR OUTPUT OF MESSAGES.
C          C6WFIL     ADAS
C          C7EMQX     ADAS
C          C6PRSL     ADAS      PREDICTS REQUESTED SPECTRUM LINES.
C
C NOTES:
C       1) THE J->J' TRANSITION INDEX IS AS FOLLOWS:
C              1 : J=L+0.5 -> J'=L'+0.5
C              2 : J=L+0.5 -> J'=L'-0.5
C              3 : J=L-0.5 -> J'=L'+0.5
C              4 : J=L-0.5 -> J'=L'-0.5
C
C AUTHOR:  JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 5183
C
C DATE:    10/11/93
C
C UNIX-IDL PORT:
C
```

```
C AUTHOR:  WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C
C DATE:    22ND MAY 1996
C
C VERSION: 1.1                          DATE: 22-05-96
C MODIFIED: WILLIAM OSBORN
C               - FIRST VERSION. COPIED FROM C6EMIS. CALL TO C6EMQX
C   REPLACED BY CALL TO C7EMQX
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IZ0,          IZ1,          MXJSHL,       MXNSHL
      INTEGER             MXOBSL,       MXPRSL,       NBOT,         NGRND
      INTEGER             NL(MXOBSL),   NOLINE,       NPL(MXPRSL),  NPLINE
      INTEGER             NPU(MXPRSL),  NTOT,         NU(MXOBSL),   NUMAX
      INTEGER             NUMIN
      REAL*8              AVRGWL(MXPRSL),             DENS,         DENSZ
      REAL*8              EM,           EMISA(MXOBSL)
      REAL*8              FTHEOR(MXJSHL,(MXNSHL*(MXNSHL+1))/2)
      REAL*8              QEFF(MXPRSL),               QEX(MXNSHL)
      REAL*8              QTHEOR(MXNSHL),             QTHIN(MXNSHL)
      REAL*8              TBFM(2*MXJSHL,(MXNSHL*(MXNSHL+1))/2)
      REAL*8              TBFMM(2*MXJSHL,(MXNSHL*(MXNSHL+1))/2)
      REAL*8              TBFMP(2*MXJSHL,(MXNSHL*(MXNSHL+1))/2)
      REAL*8              TBLEMI(2*MXJSHL,2*MXNSHL-3,MXPRSL)
      REAL*8              TBLPOP(2*MXJSHL,2*MXNSHL-3,MXPRSL)
      REAL*8              TBLWLN(2*MXJSHL,2*MXNSHL-3,MXPRSL)
      REAL*8              TBQMEM(2*MXJSHL,(MXNSHL*(MXNSHL+1))/2)
      REAL*8              TBQMEP(2*MXJSHL,(MXNSHL*(MXNSHL+1))/2)
      REAL*8              TBQMIM(2*MXJSHL,(MXNSHL*(MXNSHL+1))/2)
      REAL*8              TBQMIP(2*MXJSHL,(MXNSHL*(MXNSHL+1))/2)
      REAL*8              TOTEMI(MXPRSL),             TOTPOP(MXPRSL)
```

4.40 c8chrg: Subroutine c8chrg from library adas3xx

```
      SUBROUTINE C8CHRG( SYMBD , IZD   , SYMBR , IZR   , IDZ0  ,
     &                   IRZ0  , IRZ1  , IRZ2
     &                 )
C
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: C8CHRG *********************
C
C  PURPOSE:  SETS UP NUCLEAR CHARGE OF DONOR AND NULEAR, INITIAL AND
C            FINAL CHARGES OF RECEIVER. CHECKS VALIDITY OF RECEIVER
C            CHARGES.
C
C  CALLING PROGRAM: ADAS308
C
C  INPUT : (C*2)  SYMBD   = DONOR ELEMENT SYMBOL.
C  INPUT : (I*4)  IZD     = DONOR ION CHARGE.
C  INPUT : (C*2)  SYMBR   = RECEIVER ELEMENT SYMBOL.
C  INPUT : (I*4)  IZR     = RECEIVER ION CHARGE.
C
C  OUTPUT: (I*4)  IDZ0    = DONOR NUCLEAR CHARGE.
C  OUTPUT: (I*4)  IRZ0    = RECEIVER NUCLEAR CHARGE.
C  OUTPUT: (I*4)  IRZ1    = RECEIVER ION INITIAL CHARGE.
C  OUTPUT: (I*4)  IRZ2    = RECEIVER ION FINAL CHARGE.
C
C ROUTINES:
C           ROUTINE     SOURCE      BRIEF DESCRIPTION
C           ------------------------------------------------------------
C           I4UNIT      ADAS        RETURNS UNIT NO. FOR OUTPUT MESSAGES
C           CXCHRG      ADAS        RETURNS DONOR NUCLEAR CHARGE AND
C                                   RECEIVER NULEAR, INITIAL AND FINAL
C                                   CHARGES.
C
C AUTHOR:  JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 5183
C
C DATE:    26/11/93
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      CHARACTER*2         SYMBD,       SYMBR
      INTEGER             IDZ0,        IRZ0,        IRZ1,        IRZ2
      INTEGER             IZD,         IZR
```

4.41 c8emis: Subroutine c8emis from library adas3xx

```
      SUBROUTINE C8EMIS( MXNSHL , MXOBSL , MXPRSL , IZ0    ,
     &                   IZ1    , NGRND  , NTOT   , NBOT   ,
     &                   DENSZ  , DENS   , NOLINE , NU     ,
     &                   NL     , EMISA  , NPLINE , NPU    ,
     &                   NPL    , QTHEOR , FTHEOR , TBQMEP ,
     &                   TBQMEM , TBQMIP , TBQMIM , NUMIN  ,
     &                   NUMAX  , EM     , QEX    , TOTPOP ,
     &                   TOTEMI , AVRGWL , QEFF   , TBLPOP ,
     &                   TBLEMI , TBLWLN
     &                 )
C
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: C8EMIS *********************
C
C  PURPOSE:  PREDICTS THE L-RESOLVED EMISSIVITY FOR REQUESTED
C            TRANSITIONS.
C
C  CALLING PROGRAM: ADAS308
C
C  INPUT : (I*4)  MXNSHL   = MAXIMUM VALUE OF PRINCIPAL QUANTUM NUMBER.
C  INPUT : (I*4)  MXOBSL   = MAXIMUM NUMBER OF OBSERVED SPECTRUM LINES.
C  INPUT : (I*4)  MXPRSL   = MAXIMUM NUMBER OF SPECTRUM LINES TO
C                            PREDICT.
C  INPUT : (I*4)  IZ0      = NUCLEAR CHARGE.
C  INPUT : (I*4)  IZ1      = ION CHARGE.
C  INPUT : (I*4)  NGRND    = PRINCIPAL QUANTUM NUMBER OF GROUND STATE.
C  INPUT : (I*4)  NTOT     = PRINCIPAL QUANTUM NUMBER OF HIGHEST BOUND
C                            STATE.
C  INPUT : (I*4)  NBOT     = MINIMUM PRINCIPAL QUANTUM NUMBER.
C  INPUT : (R*8)  DENSZ    = PLASMA ION DENSITY.
C                            UNITS: CM-3
C  INPUT : (R*8)  DENS     = ELECTRON DENSITY.
C                            UNITS: CM-3
C  INPUT : (I*4)  NOLINE   = NUMBER OF OBSERVED SPECTRUM LINES.
C  INPUT : (I*4)  NU()     = LIST OF UPPER PRINCIPAL QUANTUM NUMBERS OF
C                            OBSERVED SPECTRUM LINES.
C                            DIMENSION: SPECTRUM LINE INDEX.
C  INPUT : (I*4)  NL()     = LIST OF LOWER PRINCIPAL QUANTUM NUMBERS OF
C                            OBSERVED SPECTRUM LINES.
C                            DIMENSION: SPECTRUM LINE INDEX.
C  INPUT : (R*8)  EMISA()  = LIST OF EMISSIVITIES OF OBSERVED SPECTRUM
C                            LINES.
C                            DIMENSION: SPECTRUM LINE INDEX.
C  INPUT : (I*4)  NPLINES  = NUMBER OF SPECTRUM LINES TO PREDICT.
C  INPUT : (I*4)  NPU()    = LIST OF UPPER PRINCIPAL QUANTUM NUMBERS OF
C                            SPECTRUM LINES TO PREDICT.
C                            DIMENSION: PREDICTED LINE INDEX.
C  INPUT : (I*4)  NPL()    = LIST OF LOWER PRINCIPAL QUANTUM NUMBERS OF
C                            SPECTRUM LINES TO PREDICT.
C                            DIMENSION: PREDICTED LINE INDEX.
C  INPUT : (R*8)  QTHEOR() = MEAN CHARGE EXCHANGE OR EXCITATION RATE
C                            COEFFICIENTS FOR N-LEVELS AVERAGED OVER
C                            BEAM FRACTIONS.
C                            UNITS: CM3 SEC-1
C                            DIMENSION: REFERENCED BY N QUANTUM NUMBER.
C  INPUT : (R*8)  FTHEOR() = MEAN CHARGE EXCHANGE OR EXCITATION RATE
C                            FOR NL-LEVELS AS A FRACTION OF
```

```
C                               CORRESPONDING N-LEVEL.
C                               DIMENSION: REFERENCED BY FUNC I4IDFL(N,L).
C  INPUT : (R*8)  TBQMEP() = ELECTRON RATE COEFFT. FOR NL->NL+1.
C                               INDEX FOR NL->NL+1 TRANSITION GIVEN BY
C                               I4IDFL(N,L).
C                               DIMENSION: REFERENCED BY FUNC I4IDFL(N,L).
C  INPUT : (R*8)  TBQMEM() = ELECTRON RATE COEFFT. FOR NL+1->NL.
C                               INDEX FOR NL+1->NL TRANSITION GIVEN BY
C                               I4IDFL(N,L+1).
C                               DIMENSION: REFERENCED BY FUNC I4IDFL(N,L).
C  INPUT : (R*8)  TBQMIP() = POSITIVE ION RATE COEFFT. FOR NL->NL+1.
C                               INDEX FOR NL->NL+1 TRANSITION GIVEN BY
C                               I4IDFL(N,L).
C                               DIMENSION: REFERENCED BY FUNC I4IDFL(N,L).
C  INPUT : (R*8)  TBQMIM() = POSITIVE ION RATE COEFFT. FOR NL+1->NL.
C                               INDEX FOR NL+1->NL TRANSITION GIVEN BY
C                               I4IDFL(N,L+1).
C                               DIMENSION: REFERENCED BY FUNC I4IDFL(N,L).
C
C  OUTPUT: (I*4)  NUMIN    = MINIMUM UPPER PRINCIPAL QUANTUM NUMBER FOR
C                               OBSERVED SPECTRUM LINES.
C  OUTPUT: (I*4)  NUMAX    = MAXIMUM UPPER PRINCIPAL QUANTUM NUMBER FOR
C                               OBSERVED SPECTRUM LINES.
C  OUTPUT: (R*8)  EM       = EMMISSION MEASURE.
C  OUTPUT: (R*8)  QEX()    =
C                               DIMENSION: MXNSHL.
C  OUTPUT: (R*8)  TOTPOP() = TOTAL COLLISION POP. FOR PREDICTED SPECTRUM
C                               LINE.
C                               UNITS: CM-2
C                               DIMENSION: PREDICTED LINE INDEX.
C  OUTPUT: (R*8)  TOTEMI() = TOTAL COLLISION EMISSIVITIES FOR PREDICTED
C                               SPECTRUM LINE.
C                               UNITS: PH CM-2 SEC-1
C                               DIMENSION: PREDICTED LINE INDEX.
C  OUTPUT: (R*8)  AVRGWL() = AVERAGE AIR WAVELENGTH FOR PREDICTED
C                               SPECTRUM LINE.
C                               UNITS: A
C                               DIMENSION: PREDICTED LINE INDEX.
C  OUTPUT: (R*8)  QEFF()   = EFF. RATE COEFFICIENT FOR PREDICTED
C                               SPECTRUM LINE.
C                               UNITS:
C                               DIMENSION: PREDICTED LINE INDEX.
C  OUTPUT: (R*8)  TBLPOP(,)= TABLE OF COLLISION POP. FOR PREDICTED
C                               SPECTRUM LINE.
C                               UNITS: CM-2
C                               1ST DIMENSION: PREDICTED LINE INDEX.
C                               2ND DIMENSION: REFERENCED BY FUNC I4IDLI().
C  OUTPUT: (R*8)  TBLEMI(,)= TABLE OF COLLISION EMISSIVITIES FOR
C                               PREDICTED SPECTRUM LINE.
C                               UNITS: PH CM-2 SEC-1
C                               1ST DIMENSION: PREDICTED LINE INDEX.
C                               2ND DIMENSION: REFERENCED BY FUNC I4IDLI().
C  OUTPUT: (R*8)  TBLWLN(,)= TABLE OF WAVELENGTHS FOR PREDICTED SPECTRUM
C                               LINE.
C                               UNITS: A
C                               1ST DIMENSION: PREDICTED LINE INDEX.
C                               2ND DIMENSION: REFERENCED BY FUNC I4IDLI().
C
C  PARAM : (I*4)  MXN      = MXNSHL.
C  PARAM : (I*4)  MXOB     = MXOBSL.
C
```

```
C          (I*4)  NREP     =
C          (I*4)  IC       = LOOP INDEX.
C
C          (I*4)  ICREP()  =
C                            DIMENSION: MXOB.
C
C          (R*8)  WHIGH()  =
C                            DIMENSION: REFERENCED BY L+1.
C          (R*8)  WLOW(,)  =
C                            1ST DIMENSION: REFERENCED BY I4IDFL(N,L).
C                            2ND DIMENSION: REFERENCED BY L+1.
C
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4UNIT     ADAS      RETURNS UNIT NO. FOR OUTPUT OF MESSAGES.
C          CXWFIL     ADAS
C          CXEMQX
C          CXPRSL     ADAS      PREDICTS REQUESTED SPECTRUM LINES.
C
C AUTHOR:  JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 5183
C
C DATE:    15/10/93
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IZ0,          IZ1,          MXNSHL,      MXOBSL
      INTEGER             MXPRSL,       NBOT,         NGRND
      INTEGER             NL(MXOBSL),   NOLINE,       NPL(MXPRSL), NPLINE
      INTEGER             NPU(MXPRSL),  NTOT,         NU(MXOBSL),  NUMAX
      INTEGER             NUMIN
      REAL*8              AVRGWL(MXPRSL),             DENS,        DENSZ
      REAL*8              EM,           EMISA(MXOBSL)
      REAL*8              FTHEOR((MXNSHL*(MXNSHL+1))/2)
      REAL*8              QEFF(MXPRSL),               QEX(MXNSHL)
      REAL*8              QTHEOR(MXNSHL)
      REAL*8              TBLEMI(MXPRSL,2*MXNSHL-3)
      REAL*8              TBLPOP(MXPRSL,2*MXNSHL-3)
      REAL*8              TBLWLN(MXPRSL,2*MXNSHL-3)
      REAL*8              TBQMEM((MXNSHL*(MXNSHL+1))/2)
      REAL*8              TBQMEP((MXNSHL*(MXNSHL+1))/2)
      REAL*8              TBQMIM((MXNSHL*(MXNSHL+1))/2)
      REAL*8              TBQMIP((MXNSHL*(MXNSHL+1))/2)
      REAL*8              TOTEMI(MXPRSL),             TOTPOP(MXPRSL)
```

4.42 c8prsl: Subroutine c8prsl from library adas3xx

```
      SUBROUTINE C8PRSL( MXNSHL , MXPRSL , IZ0    , IZ1    ,
     &                   NPLINE , NPU    , NPL    , NUMAX  ,
     &                   WHIGH  , WLOW   , EM     , QEX    ,
     &                   TOTPOP , TOTEMI , AVRGWL , QEFF   ,
     &                   TBLPOP , TBLEMI , TBLWLN
     &                 )
C
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: C8PRSL *********************
C
C  PURPOSE:  FILLS TABLES FOR REQUESTED PREDICTIONS OF SPECTRUM LINES.
C
C  CALLING PROGRAM: C8EMIS
C
C  INPUT : (I*4)  MXNSHL   = MAXIMUM VALUE OF PRINCIPAL QUANTUM NUMBER.
C  INPUT : (I*4)  MXPRSL   = MAXIMUM NUMBER OF SPECTRUM LINES TO
C                            PREDICT.
C  INPUT : (R*8)  IZ0      = NUCLEAR CHARGE.
C  INPUT : (R*8)  IZ1      = ION CHARGE.
C  INPUT : (I*4)  NPLINE   = NUMBER OF SPECTRUM LINES TO PREDICT.
C  INPUT : (I*4)  NPU()    = LIST OF UPPER PRINCIPAL QUANTUM NUMBERS OF
C                            SPECTRUM LINES TO PREDICT.
C                            DIMENSION: PREDICTED LINE INDEX.
C  INPUT : (I*4)  NPL()    = LIST OF LOWER PRINCIPAL QUANTUM NUMBERS OF
C                            SPECTRUM LINES TO PREDICT.
C                            DIMENSION: PREDICTED LINE INDEX.
C  INPUT : (I*4)  NUMAX    = MAXIMUM UPPER PRINCIPAL QUANTUM NUMBER FOR
C                            OBSERVED SPECTRUM LINES.
C  INPUT : (R*8)  WHIGH()  =
C                            DIMENSION: REFERENCED BY FUNC I4IDFL(N,L).
C  INPUT : (R*8)  WLOW(,)  =
C                            1ST DIMENSION: REFERENCED BY I4IDFL(N,L).
C                            2ND DIMENSION: REFERENCED BY L+1.
C  INPUT : (R*8)  EM       = EMMISION MEASURE.
C  INPUT : (R*8)  QEX()    =
C                            DIMENSION: MXNSHL.
C
C  OUTPUT: (R*8)  TOTPOP() = TOTAL COLLISION POP. FOR PREDICTED SPECTRUM
C                            LINE.
C                            UNITS: CM-2
C                            DIMENSION: PREDICTED LINE INDEX.
C  OUTPUT: (R*8)  TOTEMI() = TOTAL COLLISION EMISSIVITIES FOR PREDICTED
C                            SPECTRUM LINE.
C                            UNITS: PH CM-2 SEC-1
C                            DIMENSION: PREDICTED LINE INDEX.
C  OUTPUT: (R*8)  AVRGWL() = AVERAGE AIR WAVELENGTH FOR PREDICTED
C                            SPECTRUM LINE.
C                            UNITS: A
C                            DIMENSION: PREDICTED LINE INDEX.
C  OUTPUT: (R*8)  QEFF()   = EFF. RATE COEFFICIENT FOR PREDICTED
C                            SPECTRUM LINE.
C                            UNITS:
C                            DIMENSION: PREDICTED LINE INDEX.
C  OUTPUT: (R*8)  TBLPOP(,)= TABLE OF COLLISION POP. FOR PREDICTED
C                            SPECTRUM LINE.
C                            UNITS: CM-2
C                            1ST DIMENSION: PREDICTED LINE INDEX.
```

```
C                              2ND DIMENSION: REFERENCED BY FUNC I4IDLI().
C  OUTPUT: (R*8)  TBLEMI(,)= TABLE OF COLLISION EMISSIVITIES FOR
C                              PREDICTED SPECTRUM LINE.
C                              UNITS: PH CM-2 SEC-1
C                              1ST DIMENSION: PREDICTED LINE INDEX.
C                              2ND DIMENSION: REFERENCED BY FUNC I4IDLI().
C  OUTPUT: (R*8)  TBLWLN(,)= TABLE OF WAVELENGTHS FOR PREDICTED SPECTRUM
C                              LINE.
C                              UNITS: A
C                              1ST DIMENSION: PREDICTED LINE INDEX.
C                              2ND DIMENSION: REFERENCED BY FUNC I4IDLI().
C
C  PARAM : (I*4)  C1       = PRECISION AIR WAVELENGTH PARAM.
C  PARAM : (I*4)  C2       = PRECISION AIR WAVELENGTH PARAM.
C  PARAM : (I*4)  C3       = PRECISION AIR WAVELENGTH PARAM.
C  PARAM : (I*4)  C4       = PRECISION AIR WAVELENGTH PARAM.
C  PARAM : (I*4)  C5       = PRECISION AIR WAVELENGTH PARAM.
C  PARAM : (I*4)  RZ       = PRECISION AIR WAVELENGTH PARAM.
C
C          (I*4)  IN       = LOOP INDEX FOR SPECTRUM LINES.
C          (I*4)  N        = PRINCIPAL QUANTUM NUMBER OF INITIAL STATE.
C          (I*4)  L        = LOOP INDEX FOR ORBITAL QUANTUM NUMBER OF
C                            INITIAL STATE.
C          (I*4)  N1       = PRINCIPAL QUANTUM NUMBER OF FINAL STATE.
C          (I*4)  L1       = LOOP INDEX FOR ORBITAL QUANTUM NUMBER OF
C                            FINAL STATE.
C          (I*4)  NP       = LOOP INDEX FOR PRINCIPAL QUANTUM NUMBER.
C          (I*4)  IDL      = ARRAY INDEX.
C          (I*4)  ID       = ARRAY INDEX.
C
C          (R*8)  Z1       = REAL VALUE = IZ1.
C          (R*8)  T1       = COL. POP. FOR PREDICTED SPECTRUM LINE.
C                            UNITS: CM-2
C          (R*8)  T2       = COL. EMIS. FOR PREDICTED SPECTRUM LINE.
C                            UNITS: PH CM-2 SEC-1
C          (R*8)  E0       = BINDING ENERGY
C                            UNITS: RYD
C          (R*8)  E10      = BINDING ENERGY
C                            UNITS: RYD
C          (R*8)  DELTA    =
C          (R*8)  SIG2     =
C          (R*8)  RF       =
C          (R*8)  WAVAIR   = WAVELENGTH FOR PREDICTED SPECTRUM LINE.
C                            UNITS: A
C          (R*8)  SUM1     = SUM OF COL. POP. FOR PREDICTED LINE.
C                            UNITS: CM-2
C          (R*8)  SUM2     = SUM OF COL. EMIS. FOR PREDICTED LINE.
C                            UNITS: PH CM-2 SEC-1
C          (R*8)  SUM3     = SUM OF WAVELENGTHS FOR PREDICTED LINE.
C                            UNITS: A
C
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4UNIT     ADAS      RETURNS UNIT NO. FOR OUTPUT OF MESSAGES.
C          I4IDFL     ADAS      RETURNS UNIQUE INDEX GIVEN QUANTUM
C                               NUMBERS N AND L.
C          I4IDLI     ADAS      RETURNS INDEX FOR PREDICTED SPECTRUM
C                               LINE TABLES.
C          R8ATAB     ADAS      RETURNS HYDRONIC L-RESOLVED A-VALUES.
```

```
C                               IF INPUT QUANTUM NUMBERS ARE INVALID
C                               THEN RETURNS ZERO.
C          R8CONST    ADAS      RETURNS FUNDAMENTAL ATOMIC CONSTANTS.
C
C
C AUTHOR:  JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 5183
C
C DATE:    14/10/93
C
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C           ADDED ERROR MESSAGE FOR NUMERICAL ERRORS INTRODUCED
C           BY BAD VALUES FOR THE OBSERVED SPECTRUM LINES
C           ALSO GENERAL UNIX PORT.
C
C DATE:     20/6/95
C
C VERSION: 1.2
C MODIFIED: Martin O'Mullane
C           - Corrected the vacuum to air conversion. Parameter C4 of the
C             expression for the refractive index of air should be 146.0
C             and not 176.0. (see Astrophysical Quantities section 54)
C           - Introduced R8CONST to return fundamental atomic constants.
C             The mass dependent rydberg constant is returned if set
C             by XXRAMS in the main program. As a consequence RZ has
C             been removed from the parameter statement.
C
C DATE:     08/04/99
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IZ0,         IZ1,         MXNSHL,      MXPRSL
      INTEGER             NPL(MXPRSL), NPLINE,      NPU(MXPRSL), NUMAX
      REAL*8              AVRGWL(MXPRSL),           EM
      REAL*8              QEFF(MXPRSL),             QEX(MXNSHL)
      REAL*8              TBLEMI(MXPRSL,2*MXNSHL-3)
      REAL*8              TBLPOP(MXPRSL,2*MXNSHL-3)
      REAL*8              TBLWLN(MXPRSL,2*MXNSHL-3)
      REAL*8              TOTEMI(MXPRSL),           TOTPOP(MXPRSL)
      REAL*8              WHIGH((MXNSHL*(MXNSHL+1))/2)
      REAL*8              WLOW((MXNSHL*(MXNSHL+1))/2,MXNSHL)
```

4.43 c8tbqm: Subroutine c8tbqm from library adas3xx

```
      SUBROUTINE C8TBQM( MXNSHL , IZ0    , IZ1    , NBOT   ,
     &                   NTOP   , TEV    , DENS   , ZP     ,
     &                   TPV    , EMP    , TBLF   , TBQMEP ,
     &                   TBQMEM , TBQMIP , TBQMIM
     &                 )
C
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: C8TBQM *********************
C
C  PURPOSE:  SETS UP TABLES OF ELECTRON AND POSITIVE ION IMPACT RATE
C            COEFFICIENTS BETWEEN NEARLY DEGENERATE L STATES OF THE
C            SAME N FOR HYDROGENIC IONS.
C
C  CALLING PROGRAM: ADAS308
C
C  INPUT : (I*4)  MXNSHL   = MAXIMUM VALUE OF PRINCIPAL QUANTUM NUMBER.
C  INPUT : (I*4)  IZ0      = NUCLEAR CHARGE OF TARGET ION.
C  INPUT : (I*4)  IZ1      = ION CHARGE.
C  INPUT : (I*4)  NBOT     = MINIMUM PRINCIPAL QUANTUM NUMBER.
C  INPUT : (I*4)  NTOP     = MAXIMUM PRINCIPAL QUANTUM NUMBER.
C  INPUT : (R*8)  TEV      = ELECTRON TEMPERATURE.
C                            UNITS: EV
C  INPUT : (R*8)  DENS     = ELECTRON DENSITY.
C                            UNITS: CM-3
C  INPUT : (R*8)  ZP       = CHARGE OF COLLIDING POSITIVE ION.
C  INPUT : (R*8)  TPV      = TEMPERATURE (COLLIDING POSITIVE ION
C                            DISTRIBUTION).
C                            UNITS: EV
C  INPUT : (R*8)  EMP      = REDUCED MASS FOR COLLIDING POSITIVE ION.
C                            UNITS: ELECTRON MASSES
C  INPUT : (R*8)  ZP       = CHARGE OF COLLIDING POSITIVE ION.
C  INPUT : (R*8)  TPV      = POSITIVE ION TEMPERATURE.
C                            UNITS: EV
C  INPUT : (R*8)  EMP      = REDUCED MASS FOR COLLIDING POSITIVE ION.
C                            UNITS: ELECTRON MASSES
C  INPUT : (R*8)  TBLF()   = TABLE OF RADIATIVE LIFETIMES.
C                            UNITS: SECS
C                            DIMENSION: REFERENCED BY FUNC I4IDFL(N,L).
C
C  OUTPUT: (R*8)  TBQMEP() = ELECTRON RATE COEFFT. FOR NL->NL+1.
C                            INDEX FOR NL->NL+1 TRANSITION GIVEN BY
C                            I4IDFL(N,L).
C                            DIMENSION: REFERENCED BY FUNC I4IDFL(N,L).
C  OUTPUT: (R*8)  TBQMEM() = ELECTRON RATE COEFFT. FOR NL+1->NL.
C                            INDEX FOR NL+1->NL TRANSITION GIVEN BY
C                            I4IDFL(N,L+1).
C                            DIMENSION: REFERENCED BY FUNC I4IDFL(N,L).
C  OUTPUT: (R*8)  TBQMIP() = POSITIVE ION RATE COEFFT. FOR NL->NL+1.
C                            INDEX FOR NL->NL+1 TRANSITION GIVEN BY
C                            I4IDFL(N,L).
C                            DIMENSION: REFERENCED BY FUNC I4IDFL(N,L).
C  OUTPUT: (R*8)  TBQMIM() = POSITIVE ION RATE COEFFT. FOR NL+1->NL.
C                            INDEX FOR NL+1->NL TRANSITION GIVEN BY
C                            I4IDFL(N,L+1).
C                            DIMENSION: REFERENCED BY FUNC I4IDFL(N,L).
C
C  PARAM : (I*4)  MXJ     = MAXIMUM NUMBER OF J SUB-SHELLS.
```

```
C  PARAM : (R*8)  P1      =
C  PARAM : (R*8)  P2      =
C
C          (I*4)  NI      = VALENCE ELECTRON PRINCIPAL QUANTUM NUMBER
C                           IN STATE I.
C          (I*4)  NJ      = VALENCE ELECTRON PRINCIPAL QUANTUM NUMBER
C                           IN STATE J.
C          (I*4)  LI      = VALENCE ELECTRON ORBITAL QUANTUM NUMBER IN
C                           STATE I.
C          (I*4)  LJ      = VALENCE ELECTRON ORBITAL QUANTUM NUMBER IN
C                           STATE J.
C          (I*4)  IDLI    = TABLE INDEX.
C          (I*4)  IDLJ    = TABLE INDEX.
C          (I*4)  I       = LOOP INDEX.
C
C          (R*8)  FACE    =
C          (R*8)  FACI    =
C          (R*8)  WI      = STATISTICAL WEIGHT OF STATE I.
C          (R*8)  WJ      = STATISTICAL WEIGHT OF STATE J.
C          (R*8)  GAE     = GAMA RATE PARAMETER FOR ELECTRON COLLISIONS.
C          (R*8)  GAP     = GAMA RATE PARAMETER FOR POSITIVE ION
C                           COLLISIONS.
C          (R*8)  QEP()   = ELECTRON RATE COEFFT. FOR NLJ->NL+1J'
C                           DIMENSION: J->J' TRANSITION INDEX.
C          (R*8)  QEM()   = ELECTRON RATE COEFFT. FOR NLJ->NL-1J'
C                           DIMENSION: J->J' TRANSITION INDEX.
C          (R*8)  QIP()   = POSITIVE ION RATE COEFFT. FOR NLJ->NL+1J'
C                           DIMENSION: J->J' TRANSITION INDEX.
C          (R*8)  QIM()   = POSITIVE ION RATE COEFFT. FOR NLJ->NL-1J'
C                           DIMENSION: J->J' TRANSITION INDEX.
C
C NOTES:
C       1) THE J->J' TRANSITION INDEX IS AS FOLLOWS:
C              1 : J=L+0.5 -> J'=L'+0.5
C              2 : J=L+0.5 -> J'=L'-0.5
C              3 : J=L-0.5 -> J'=L'+0.5
C              4 : J=L-0.5 -> J'=L'-0.5
C
C       2) BEFORE CALLING C8TBQM THE LIFETIME TABLE MUST BE FILLED
C          WITH A CALL TO CXTBLF.
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4IDFL     ADAS      RETURNS UNIQUE INDEX GIVEN QUANTUM
C                               NUMBERS N AND L.
C          CXCRDG     ADAS      CALCULATES COLLISIONAL RATE COEFFICIENTS
C                               BETWEEN NEARLY DEGENERATE LEVELS OF
C                               H-, LI- OR NA-LIKE IONS.
C
C AUTHOR:  JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 5183
C
C DATE:    12/10/93
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IZ0,         IZ1,         MXNSHL,      NBOT
      INTEGER             NTOP
```

```
REAL*8              DENS,         EMP
REAL*8              TBLF((MXNSHL*(MXNSHL+1))/2)
REAL*8              TBQMEM((MXNSHL*(MXNSHL+1))/2)
REAL*8              TBQMEP((MXNSHL*(MXNSHL+1))/2)
REAL*8              TBQMIM((MXNSHL*(MXNSHL+1))/2)
REAL*8              TBQMIP((MXNSHL*(MXNSHL+1))/2),          TEV
REAL*8              TPV,          ZP
```

4.44 c9cxee: Subroutine c9cxee from library adas3xx

```
      SUBROUTINE C9CXEE( MXNENG , MXNSHL , NGRND  , NTOT   ,
     &                   NBOT   , NTOP   , IRZ0   , IRZ1   ,
     &                   RAMSNO , TEV    , TIEV   , DENS   ,
     &                   DENSZ  , ZEFF   , BMAG   , BMENG  ,
     &                   ITHEOR , IBSTAT , IEMMS  , NTU    ,
     &                   NTL    , NMINF  , NMAXF  , NENRGY ,
     &                   ENRGYA , ALPHAA , XSECNA , FRACLA ,
     &                   ERATE
     &                 )
C
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: C9CXEE *********************
C
C  PURPOSE:  CALCULATES THE L-RESOLVED EFFECTIVE EMISSIVITY RATE
C            COEFFICIENT FOR THE GIVEN TRANSITION.
C
C            IT IS APPLICABLE TO IMPURITIES IN PLASMA TRAVERSED BY
C            NEUTRAL BEAMS OF H OR HE.
C
C            THE RECOMBINED TARGET ION IS TREATED AS H-LIKE.
C
C            THE MODEL INCLUDES CAPTURE, N-N' LEVEL CASCADE, AND MIXING
C            AMONG L LEVELS OF SAME N BY COLLISIONS.
C
C            AN INTERNAL EIKONAL APPROXIMATION IS USED FOR CAPTURE FROM
C            EXCITED H OR HE STATES, ALTHOUGH NORMALLY THE EXTERNAL DATA
C            SET SHOULD BE USED.
C
C  CALLING PROGRAM: ADAS309
C
C  INPUT : (I*4)  MXNENG    = MAXIMUM NO. OF ENERGIES IN DATA SET.
C  INPUT : (I*4)  MXNSHL    = MAXIMUM NUMBER OF N SHELLS.
C  INPUT : (I*4)  NGRND     = PRINCIPAL QUANTUM NUMBER OF GROUND STATE.
C  INPUT : (I*4)  NTOT      = PRINCIPAL QUANTUM NUMBER OF HIGHEST BOUND
C                             STATE.
C  INPUT : (I*4)  NBOT      = MINIMUM PRINCIPAL QUANTUM NUMBER FOR
C                             RATE TABLES.
C  INPUT : (I*4)  NTOP      = MAXIMUM PRINCIPAL QUANTUM NUMBER FOR
C                             RATE TABLES.
C  INPUT : (I*4)  IRZ0      = RECEIVER NUCLEAR CHARGE.
C  INPUT : (I*4)  IRZ1      = RECEIVER ION INITIAL CHARGE.
C  INPUT : (R*8)  RAMSNO    = RECEIVER ATOMIC MASS.
C  INPUT : (R*8)  TEV       = ELECTRON TEMPERATURE.
C                             UNITS: EV
C  INPUT : (R*8)  TIEV      = ION TEMPERATURE.
C                             UNITS: EV
C  INPUT : (R*8)  DENS      = ELECTRON DENSITY.
C                             UNITS: CM-3
C  INPUT : (R*8)  DENSZ     = PLASMA ION DENSITY.
C                             UNITS: CM-3
C  INPUT : (R*8)  ZEFF      = EFFECTIVE ION CHARGE.
C  INPUT : (R*8)  BMAG      = PLASMA MAGNETIC INDUCTION.
C                             UNITS: TESLA
C  INPUT : (R*8)  BMENG     = BEAM ENERGY.
C                             UNITS: EV/AMU
C  INPUT : (I*4)  ITHEOR    = CHARGE EXCHANGE MODEL OPTION.
C                             1 => USE INPUT DATA SET.
```

```
C                                2 => USE EIKONAL MODEL.
C  INPUT : (I*4)  IBSTAT    = DONOR STATE FOR EIKONAL MODEL.
C                                1 => H(1S)
C                                2 => H(2S)
C                                3 => H(2P)
C                                4 => HE(1S2)
C                                5 => HE(1S2S)
C  INPUT : (I*4)  IEMMS      = EMISSION MEASURE MODEL OPTION.
C                                1 => CHARGE EXCHANGE.
C                                2 => ELECTRON IMPACT EXCITATION.
C  INPUT : (I*4)  NTL        = LOWER PRINCIPAL QUANTUM NUMBER OF
C                                TRANSITION.
C  INPUT : (I*4)  NTU        = UPPER PRINCIPAL QUANTUM NUMBER OF
C                                TRANSITION.
C  INPUT : (I*4)  NMINF      = LOWEST N-SHELL FOR WHICH DATA READ.
C  INPUT : (I*4)  NMAXF      = HIGHEST N-SHELL FOR WHICH DATA READ.
C  INPUT : (I*4)  NENRGY     = NUMBER OF ENERGIES READ FROM DATA SET.
C  INPUT : (R*8)  ENRGYA()   = COLLISION ENERGIES READ FROM INPUT DATA
C                                SET.
C                                UNITS: EV/AMU
C                                DIMENSION: ENERGY INDEX
C  INPUT : (R*8)  ALPHAA()   = EXTRAPOLATION PARAMETER ALPHA READ FROM
C                                INPUT DATA SET.
C                                DIMENSION: ENERGY INDEX
C  INPUT : (R*8)  XSECNA(,) = N-RESOLVED CHARGE EXCHANGE CROSS-SECTIONS
C                                READ FROM INPUT DATA SET.
C                                UNITS: CM2
C                                1ST DIMENSION: ENERGY INDEX
C                                2ND DIMENSION: N-SHELL
C  INPUT : (R*8)  FRACLA(,) = L-RESOLVED CHARGE EXCHANGE CROSS-SECTIONS.
C                                AFTER CXDATA: ABSOLUTE VALUES (CM2).
C                                AFTER CXFRAC: FRACTION OF N-RESOLVED
C                                              DATA.
C                                1ST DIMENSION: ENERGY INDEX
C                                2ND DIMENSION: INDEXED BY I4IDFL(N,L)
C
C  OUTPUT: (R*8)  ERATE      = EFFECTIVE EMISSIVITY RATE COEFFICIENT FOR
C                                REQUESTED TRANSITION
C                                SPECTRUM LINE.
C                                UNITS: CM3 SEC-1
C
C  PARAM : (I*4)  MXN        = MXNSHL.
C  PARAM : (I*4)  MXJSHL     = MAXIMUM NUMBER OF J SUB-SHELLS.
C  PARAM : (I*4)  MXBEAM     = MAXIMUM NUMBER OF BEAM COMPONENTS.
C  PARAM : (I*4)  MXOBSL     = MAXIMUM NUMBER OF OBSERVED SPECTRUM
C                                LINES.
C
C  PARAM : (I*4)  MXPRSL     = MAXIMUM NUMBER OF SPECTRUM LINES TO
C                                PREDICT.
C
C  PARAM : (R*8)  EMP        = REDUCED MASS FOR POSITIVE ION.
C                                UNITS: ELECTRON MASSES
C
C          (I*4)  NBEAM      = NUMBER OF BEAM ENERGIES.
C          (I*4)  NOLINE     = NUMBER OF OBSERVED SPECTRUM LINES.
C          (I*4)  NPLINE     = NUMBER OF SPECTRUM LINES TO PREDICT.
C          (I*4)  NUMIN      = MINIMUM UPPER PRINCIPAL QUANTUM NUMBER
C                                FOR OBSERVED SPECTRUM LINES.
C          (I*4)  NUMAX      = MAXIMUM UPPER PRINCIPAL QUANTUM NUMBER
C                                FOR OBSERVED SPECTRUM LINES.
C
```

```
C          (R*8)  EM         = EMMISSION MEASURE.
C                               UNITS: CM-5
C
C          (I*4)  NL()       = LIST OF LOWER PRINCIPAL QUANTUM NUMBERS
C                               OF OBSERVED SPECTRUM LINES.
C                               DIMENSION: SPECTRUM LINE INDEX.
C          (I*4)  NU()       = LIST OF UPPER PRINCIPAL QUANTUM NUMBERS
C                               OF OBSERVED SPECTRUM LINES.
C                               DIMENSION: SPECTRUM LINE INDEX.
C          (I*4)  NPL()      = LIST OF LOWER PRINCIPAL QUANTUM NUMBERS
C                               OF SPECTRUM LINES TO PREDICT.
C                               DIMENSION: SPECTRUM LINE INDEX.
C          (I*4)  NPU()      = LIST OF UPPER PRINCIPAL QUANTUM NUMBERS
C                               OF SPECTRUM LINES TO PREDICT.
C                               DIMENSION: SPECTRUM LINE INDEX.
C
C          (R*8)  BMFRA()    = BEAM COMPONENT FRACTIONS.
C                               DIMENSION: COMPONENT INDEX.
C          (R*8)  BMENA()    = BEAM ENERGY COMPONENTS.
C                               UNITS: EV/AMU
C          (R*8)  EMISA()    = LIST OF EMISSIVITIES OF OBSERVED SPECTRUM
C                               LINES.
C                               UNITS: PH CM-2 SEC-1
C                               DIMENSION: SPECTRUM LINE INDEX.
C          (R*8)  TBLF()     = TABLE OF RADIATIVE LIFETIMES.
C                               UNITS: SECS
C                               DIMENSION: REFERENCED BY I4IDFL(N,L).
C          (R*8)  TBQEX()    = MEAN EXCITATION RATE COEFFICIENTS FOR
C                               NL-LEVELS AVERAGED OVER BEAM FRACTIONS.
C                               UNITS: CM3 SEC-1
C                               DIMENSION: REFERENCED BY I4IDFL(N,L).
C          (R*8)  QTHEX()    = MEAN EXCITATION RATE COEFFICIENTS FOR
C                               N-LEVELS AVERAGED OVER BEAM FRACTIONS.
C                               UNITS: CM3 SEC-1
C                               DIMENSION: N SHELL INDEX.
C          (R*8)  FTHEX()    = FRACTION OF N-LEVEL MEAN EXCITATION RATE
C                               COEFFICIENTS IN NL-LEVEL.
C                               DIMENSION: REFERENCED BY I4IDFL(N,L).
C          (R*8)  QTHCH()    = MEAN CHARGE EXCHANGE COEFFICIENTS FOR
C                               N-LEVELS AVERAGED OVER BEAM FRACTIONS.
C                               UNITS: CM3 SEC-1
C                               DIMENSION: N SHELL INDEX.
C          (R*8)  FTHCH()    = FRACTION OF N-LEVEL MEAN CHARGE EXCHANGE
C                               COEFFICIENTS IN NL-LEVEL.
C                               DIMENSION: REFERENCED BY I4IDFL(N,L).
C          (R*8)  TBQMEP()   = ELECTRON COLLISIONAL RATE COEFFT. FOR
C                               NL->NL+1.
C                               DIMENSION: REFERENCED BY I4IDFL(N,L).
C          (R*8)  TBQMEM()   = ELECTRON COLLISIONAL RATE COEFFT. FOR
C                               NL->NL-1.
C                               DIMENSION: REFERENCED BY I4IDFL(N,L).
C          (R*8)  TBQMIP()   = POSITIVE ION COLLISIONAL RATE COEFFT. FOR
C                               NL->NL+1.
C                               DIMENSION: REFERENCED BY I4IDFL(N,L).
C          (R*8)  TBQMIM()   = POSITIVE ION COLLISIONAL RATE COEFFT. FOR
C                               NL->NL-1.
C                               DIMENSION: REFERENCED BY I4IDFL(N,L).
C          (R*8)  QEX()      =
C                               DIMENSION: N SHELL INDEX.
C          (R*8)  TOTPOP()   = TOTAL COLLISION POP. FOR PREDICTED
C                               SPECTRUM LINE.
```

```
C                               UNITS: CM-2
C                               DIMENSION: PREDICTED LINE INDEX.
C          (R*8)  TOTEMI()   = TOTAL COLLISION EMISSIVITIES FOR PREDICTED
C                               SPECTRUM LINE.
C                               UNITS: PH CM-2 SEC-1
C                               DIMENSION: PREDICTED LINE INDEX.
C          (R*8)  AVRGWL()   = AVERAGE AIR WAVELENGTH FOR PREDICTED
C                               SPECTRUM LINE.
C                               UNITS: A
C                               DIMENSION: PREDICTED LINE INDEX.
C          (R*8)  QEFF()     = EFF. RATE COEFFICIENT FOR PREDICTED
C                               SPECTRUM LINE.
C                               UNITS: CM3 SEC-1
C                               DIMENSION: PREDICTED LINE INDEX.
C
C          (R*8)  TBLPOP(,) = TABLE OF COLLISION POP. FOR PREDICTED
C                               SPECTRUM LINE.
C                               UNITS: CM-2
C                               1ST DIMENSION: PREDICTED LINE INDEX.
C                               2ND DIMENSION: REFERENCED BY I4IDLI().
C          (R*8)  TBLEMI(,) = TABLE OF COLLISION EMISSIVITIES FOR
C                               PREDICTED SPECTRUM LINE.
C                               UNITS: PH CM-2 SEC-1
C                               1ST DIMENSION: PREDICTED LINE INDEX.
C                               2ND DIMENSION: REFERENCED BY I4IDLI().
C          (R*8)  TBLWLN(,) = TABLE OF WAVELENGTHS FOR PREDICTED
C                               SPECTRUM LINE.
C                               UNITS: A
C                               1ST DIMENSION: PREDICTED LINE INDEX.
C                               2ND DIMENSION: REFERENCED BY I4IDLI().
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4UNIT     ADAS      RETURNS UNIT NO. FOR OUTPUT OF MESSAGES.
C          CXTBLF     ADAS      FILLS L-RESOLVED RADIATIVE LIFETIME
C                               TABLE.
C          CXTBEX     ADAS      FILLS N AND L-RESOLVED ELECTRON IMPACT
C                               EXCITATION RATE TABLES.
C          CXQEIK     ADAS      FILLS N AND L-RESOLVED CHARGE EXCHANGE
C                               RATE TABLES USING EIKONAL APPROXIMATION.
C          CXQXCH     ADAS      FILLS N AND L-RESOLVED CHARGE EXCHANGE
C                               RATE TABLES USING INPUT DATA SET.
C          C8TBQM     ADAS      FILLS N AND L-RESOLVED COLLISIONAL RATE
C                               TABLES.
C          C9EMIS     ADAS      PREDICTS THE L-RESOLVED EMISSIVITY FOR
C                               REQUESTED TRANSITIONS.
C
C AUTHOR:  JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/87
C          JET EXT. 5183
C
C DATE:    03/12/93
C
C UNIX PORT: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C  DATE:   10/07/95 VERSION 1.1
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IBSTAT,      IEMMS,       IRZ0,        IRZ1
```

```
INTEGER             ITHEOR,      MXNENG,      MXNSHL,      NBOT
INTEGER             NENRGY,      NGRND,       NMAXF,       NMINF
INTEGER             NTL,         NTOP,        NTOT,        NTU
REAL*8              ALPHAA(MXNENG),           BMAG,        BMENG
REAL*8              DENS,        DENSZ,       ENRGYA(MXNENG)
REAL*8              ERATE
REAL*8              FRACLA(MXNENG,(MXNSHL*(MXNSHL+1))/2),  RAMSNO
REAL*8              TEV,         TIEV
REAL*8              XSECNA(MXNENG,MXNSHL),    ZEFF
```

4.45 c9emis: Subroutine c9emis from library adas3xx

```
      SUBROUTINE C9EMIS( MXNSHL , MXOBSL , MXPRSL , IZ0    ,
     &                   IZ1    , NGRND  , NTOT   , NBOT   ,
     &                   DENSZ  , DENS   , NOLINE , NU     ,
     &                   NL     , EMISA  , NPLINE , NPU    ,
     &                   NPL    , QTHEOR , FTHEOR , TBQMEP ,
     &                   TBQMEM , TBQMIP , TBQMIM , NUMIN  ,
     &                   NUMAX  , EM     , QEX    , TOTPOP ,
     &                   TOTEMI , AVRGWL , QEFF   , TBLPOP ,
     &                   TBLEMI , TBLWLN
     &                 )
C
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: C9EMIS *********************
C
C  PURPOSE:  PREDICTS THE L-RESOLVED EMISSIVITY FOR REQUESTED
C            TRANSITIONS.
C
C  CALLING PROGRAM: C9CXEE
C
C  INPUT : (I*4)  MXNSHL   = MAXIMUM VALUE OF PRINCIPAL QUANTUM NUMBER.
C  INPUT : (I*4)  MXOBSL   = MAXIMUM NUMBER OF OBSERVED SPECTRUM LINES.
C  INPUT : (I*4)  MXPRSL   = MAXIMUM NUMBER OF SPECTRUM LINES TO
C                            PREDICT.
C  INPUT : (I*4)  IZ0      = NUCLEAR CHARGE.
C  INPUT : (I*4)  IZ1      = ION CHARGE.
C  INPUT : (I*4)  NGRND    = PRINCIPAL QUANTUM NUMBER OF GROUND STATE.
C  INPUT : (I*4)  NTOT     = PRINCIPAL QUANTUM NUMBER OF HIGHEST BOUND
C                            STATE.
C  INPUT : (I*4)  NBOT     = MINIMUM PRINCIPAL QUANTUM NUMBER.
C  INPUT : (R*8)  DENSZ    = PLASMA ION DENSITY.
C                            UNITS: CM-3
C  INPUT : (R*8)  DENS     = ELECTRON DENSITY.
C                            UNITS: CM-3
C  INPUT : (I*4)  NOLINE   = NUMBER OF OBSERVED SPECTRUM LINES.
C  INPUT : (I*4)  NU()     = LIST OF UPPER PRINCIPAL QUANTUM NUMBERS OF
C                            OBSERVED SPECTRUM LINES.
C                            DIMENSION: SPECTRUM LINE INDEX.
C  INPUT : (I*4)  NL()     = LIST OF LOWER PRINCIPAL QUANTUM NUMBERS OF
C                            OBSERVED SPECTRUM LINES.
C                            DIMENSION: SPECTRUM LINE INDEX.
C  INPUT : (R*8)  EMISA()  = LIST OF EMISSIVITIES OF OBSERVED SPECTRUM
C                            LINES.
C                            DIMENSION: SPECTRUM LINE INDEX.
C  INPUT : (I*4)  NPLINES  = NUMBER OF SPECTRUM LINES TO PREDICT.
C  INPUT : (I*4)  NPU()    = LIST OF UPPER PRINCIPAL QUANTUM NUMBERS OF
C                            SPECTRUM LINES TO PREDICT.
C                            DIMENSION: PREDICTED LINE INDEX.
C  INPUT : (I*4)  NPL()    = LIST OF LOWER PRINCIPAL QUANTUM NUMBERS OF
C                            SPECTRUM LINES TO PREDICT.
C                            DIMENSION: PREDICTED LINE INDEX.
C  INPUT : (R*8)  QTHEOR() = MEAN CHARGE EXCHANGE OR EXCITATION RATE
C                            COEFFICIENTS FOR N-LEVELS AVERAGED OVER
C                            BEAM FRACTIONS.
C                            UNITS: CM3 SEC-1
C                            DIMENSION: REFERENCED BY N QUANTUM NUMBER.
C  INPUT : (R*8)  FTHEOR() = MEAN CHARGE EXCHANGE OR EXCITATION RATE
C                            FOR NL-LEVELS AS A FRACTION OF
```

```
C                              CORRESPONDING N-LEVEL.
C                              DIMENSION: REFERENCED BY FUNC I4IDFL(N,L).
C  INPUT : (R*8)  TBQMEP() = ELECTRON RATE COEFFT. FOR NL->NL+1.
C                              INDEX FOR NL->NL+1 TRANSITION GIVEN BY
C                              I4IDFL(N,L).
C                              DIMENSION: REFERENCED BY FUNC I4IDFL(N,L).
C  INPUT : (R*8)  TBQMEM() = ELECTRON RATE COEFFT. FOR NL+1->NL.
C                              INDEX FOR NL+1->NL TRANSITION GIVEN BY
C                              I4IDFL(N,L+1).
C                              DIMENSION: REFERENCED BY FUNC I4IDFL(N,L).
C  INPUT : (R*8)  TBQMIP() = POSITIVE ION RATE COEFFT. FOR NL->NL+1.
C                              INDEX FOR NL->NL+1 TRANSITION GIVEN BY
C                              I4IDFL(N,L).
C                              DIMENSION: REFERENCED BY FUNC I4IDFL(N,L).
C  INPUT : (R*8)  TBQMIM() = POSITIVE ION RATE COEFFT. FOR NL+1->NL.
C                              INDEX FOR NL+1->NL TRANSITION GIVEN BY
C                              I4IDFL(N,L+1).
C                              DIMENSION: REFERENCED BY FUNC I4IDFL(N,L).
C
C  OUTPUT: (I*4)  NUMIN    = MINIMUM UPPER PRINCIPAL QUANTUM NUMBER FOR
C                              OBSERVED SPECTRUM LINES.
C  OUTPUT: (I*4)  NUMAX    = MAXIMUM UPPER PRINCIPAL QUANTUM NUMBER FOR
C                              OBSERVED SPECTRUM LINES.
C  OUTPUT: (R*8)  EM       = EMMISSION MEASURE.
C  OUTPUT: (R*8)  QEX()    =
C                              DIMENSION: MXNSHL.
C  OUTPUT: (R*8)  TOTPOP() = TOTAL COLLISION POP. FOR PREDICTED SPECTRUM
C                              LINE.
C                              UNITS: CM-2
C                              DIMENSION: PREDICTED LINE INDEX.
C  OUTPUT: (R*8)  TOTEMI() = TOTAL COLLISION EMISSIVITIES FOR PREDICTED
C                              SPECTRUM LINE.
C                              UNITS: PH CM-2 SEC-1
C                              DIMENSION: PREDICTED LINE INDEX.
C  OUTPUT: (R*8)  AVRGWL() = AVERAGE AIR WAVELENGTH FOR PREDICTED
C                              SPECTRUM LINE.
C                              UNITS: A
C                              DIMENSION: PREDICTED LINE INDEX.
C  OUTPUT: (R*8)  QEFF()   = EFF. RATE COEFFICIENT FOR PREDICTED
C                              SPECTRUM LINE.
C                              UNITS:
C                              DIMENSION: PREDICTED LINE INDEX.
C  OUTPUT: (R*8)  TBLPOP(,)= TABLE OF COLLISION POP. FOR PREDICTED
C                              SPECTRUM LINE.
C                              UNITS: CM-2
C                              1ST DIMENSION: PREDICTED LINE INDEX.
C                              2ND DIMENSION: REFERENCED BY FUNC I4IDLI().
C  OUTPUT: (R*8)  TBLEMI(,)= TABLE OF COLLISION EMISSIVITIES FOR
C                              PREDICTED SPECTRUM LINE.
C                              UNITS: PH CM-2 SEC-1
C                              1ST DIMENSION: PREDICTED LINE INDEX.
C                              2ND DIMENSION: REFERENCED BY FUNC I4IDLI().
C  OUTPUT: (R*8)  TBLWLN(,)= TABLE OF WAVELENGTHS FOR PREDICTED SPECTRUM
C                              LINE.
C                              UNITS: A
C                              1ST DIMENSION: PREDICTED LINE INDEX.
C                              2ND DIMENSION: REFERENCED BY FUNC I4IDLI().
C
C  PARAM : (I*4)  MXN      = MXNSHL.
C  PARAM : (I*4)  MXOB     = MXOBSL.
C
```

```
C          (I*4)  NREP     =
C          (I*4)  IC       = LOOP INDEX.
C
C          (I*4)  ICREP()  =
C                            DIMENSION: MXOB.
C
C          (R*8)  WHIGH()  =
C                            DIMENSION: REFERENCED BY L+1.
C          (R*8)  WLOW(,)  =
C                            1ST DIMENSION: REFERENCED BY I4IDFL(N,L).
C                            2ND DIMENSION: REFERENCED BY L+1.
C
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4UNIT     ADAS      RETURNS UNIT NO. FOR OUTPUT OF MESSAGES.
C          C9WFIL     ADAS
C          C9EMQX
C          CXPRSL     ADAS      PREDICTS REQUESTED SPECTRUM LINES.
C
C AUTHOR:  JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 5183
C
C DATE:    15/10/93
C
C DATE:    30/06/95 VERSION:1.1
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C - UNIX PORT FOR ADAS309, CREATED C9EMIS FROM C8EMIS
C   ONLY DIFFERENCE IS IT CALLS C9WFIL RATHER THAN C8WFIL
C
C DATE:    10/07/95 VERSION:1.2
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - CHANGED CALL FROM C8EMQX TO C9EMQX
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IZ0,          IZ1,          MXNSHL,      MXOBSL
      INTEGER             MXPRSL,       NBOT,         NGRND
      INTEGER             NL(MXOBSL),   NOLINE,       NPL(MXPRSL), NPLINE
      INTEGER             NPU(MXPRSL),  NTOT,         NU(MXOBSL),  NUMAX
      INTEGER             NUMIN
      REAL*8              AVRGWL(MXPRSL),             DENS,        DENSZ
      REAL*8              EM,           EMISA(MXOBSL)
      REAL*8              FTHEOR((MXNSHL*(MXNSHL+1))/2)
      REAL*8              QEFF(MXPRSL),               QEX(MXNSHL)
      REAL*8              QTHEOR(MXNSHL)
      REAL*8              TBLEMI(MXPRSL,2*MXNSHL-3)
      REAL*8              TBLPOP(MXPRSL,2*MXNSHL-3)
      REAL*8              TBLWLN(MXPRSL,2*MXNSHL-3)
      REAL*8              TBQMEM((MXNSHL*(MXNSHL+1))/2)
      REAL*8              TBQMEP((MXNSHL*(MXNSHL+1))/2)
      REAL*8              TBQMIM((MXNSHL*(MXNSHL+1))/2)
      REAL*8              TBQMIP((MXNSHL*(MXNSHL+1))/2)
      REAL*8              TOTEMI(MXPRSL),             TOTPOP(MXPRSL)
```

4.46 c9ispf: Subroutine c9ispf from library adas3xx

```
      SUBROUTINE C9ISPF( LPEND, IBATCH )
C
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: C9ISPF *********************
C
C  PURPOSE: PIPE COMMS WITH IDL FOR THE PROCESSING OPTIONS OF 309.
C
C  CALLING PROGRAM: ADAS309
C
C  I/O   : (L*4)   LPEND    = FLAGS IF END OF ANALYSIS REQUESTED.
C                              .TRUE.  => END ANALYSIS OF CURRENT DATA
C                                         SETS
C                              .FALSE. => CONTINUE PANALYSIS WITH CURRENT
C                                         DATA SETS
C
C
C          (I*4)   PIPEIN - PARAMETER = UNIT NUMBER FOR INPUT FROM PIPE
C          (I*4)   LOGIC  - USED TO PIPE LOGICAL VALUES
C    (I*4)   IBATCH - USED TO FLAG ACTION TO OTHER ROUTINES (
C     AS LPEND CAN NOW HAVE 3 VALUES WHICH IT IS
C                           NOT POSSIBLE TO TREAT WITH A LOGICAL
C                           VARIABLE)
C
C ROUTINES:   NONE
C
C AUTHOR:  JONATHAN NASH  (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 5183
C
C DATE:    03/12/93
C
C Modified: Tim Hammond (Tessella Support Services plc.)
C           Preliminary UNIX port - major revisions to follow
C
C Version 1.1 21-06-95
C
C Modified: Tim Hammond (Tessella Support Services plc.)
C - Removed references to ISPF panel IPAN.
C - Removed other ISPF variables and features.
C - Removed batch/no-batch flag LBTSEL
C - Added all reads from IDL (c9ispf.pro)
C
C Version 1.2 22-06-95
C
C Modified: Tim Hammond (Tessella Support Services plc.)
C               - Added pipein parameter
C
C Version 1.3 28-06-95
C
C Modified: Tim Hammond (Tessella Support Services plc.)
C               - Rewrite of the routine - it now only reads
C                 whether to carry on or not.
C   The bulk of the IDL -> FORTRAN data transfer is
C   now handled by routines under adas309b.for
C
C Version 1.4 10-07-95
C
```

```
C Modified: Tim Hammond (Tessella Support Services plc.)
C               - Added IBATCH parameter which is used to signal
C                 'Cancel', 'Run Now', 'Run in Batch' to
C                 other routines.
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IBATCH
      LOGICAL             LPEND
```

4.47 cachkb: Subroutine cachkb from library adas3xx

```
      SUBROUTINE CACHKB( IUNIT  , NBSEL  , IBSEL  ,
     &                   IZ0IN  , IZ0    ,
     &                   LOPEN  , IRCODE , DLPATH
     &                 )
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: CACHKB *********************
C
C  PURPOSE: TO CHECK THE SELECTED BLOCK (IBSEL) OF DATA  EXISTS  IN  THE
C           INPUT DATA SET AND IF SO IT REPRESENTS THE ENTERED VALUES OF
C           'IZ0IN' (NUCLEAR CHARGE OF SELECTED IONISING ION ELEMENT).
C
C           IT ALSO CLOSES THE INPUT DATA SET ALLOCATION IF OPEN.
C
C  CALLING PROGRAM: CASSZD
C
C  SUBROUTINE:
C
C  INPUT : (I*4)   IUNIT   = UNIT TO WHICH INPUT DATA SET IS ALLOCATED
C  INPUT : (I*4)   NBSEL   = TOTAL NUMBER OF DATA-BLOCKS READ FROM INPUT
C                            DATA SET.
C  INPUT : (I*4)   IBSEL   = INDEX OF DATA-BLOCK SELECTED FOR ANALYSIS
C
C  INPUT : (I*4)   IZ0IN   = REQUESTED: NUCLEAR CHARGE OF ELEMENT
C  INPUT : (I*4)   IZ0     = INPUT FILE: NUCLEAR CHARGE OF ELEMENT
C
C  I/O   : (L*4)   LOPEN   = INPUT : .TRUE.  => INPUT DATA SET OPEN.
C                                    .FALSE. => INPUT DATA SET CLOSED.
C                            OUTPUT: ALWAYS RETURNED AS .FALSE.
C  OUTPUT: (I*4)   IRCODE  = RETURN CODE FROM SUBROUTINE:
C                            0 => NO ERROR DETECTED.
C                            2 => DISCREPANCY BETWEEN REQUESTED ELEMENT
C                                 AND THAT IN INPUT DATA FILE.
C                            3 => SELECTED DATA-BLOCK  OUT OF RANGE  OR
C                                 DOES NOT EXIST.
C
C           (I*4)   I4UNIT  = FUNCTION (SEE ROUTINE SECTION BELOW)
C
CA          (c*80)  DSNAME  = DATA FILE NAME UNDER UNIX, INCLUDING PATH.
C
C ROUTINES:
C           ROUTINE    SOURCE    BRIEF DESCRIPTION
C           ------------------------------------------------------------
C           E2FILE     ADAS      OPEN DATA SET FOR SELECTED ELEMENT
C           I4UNIT     ADAS      FETCH UNIT NUMBER FOR OUTPUT OF MESSAGES
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/37
C          JET EXT. 2620
C
C DATE:    06/06/91
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 25-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - CREATED FROM E2CHKB TO WHICH IT IS ESSENTIALLY
C                 IDENTICAL EXCEPT IT CALLS CAFILE RATHER THAN E2FILE
```

```
C                 AND HAS THE EXTRA PATHNAME VARIABLE DLPATH INCLUDED
C                 IN THE ARGUMENT LIST.
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      CHARACTER*80        DLPATH
      INTEGER             IBSEL,       IRCODE,      IUNIT,       IZ0
      INTEGER             IZ0IN,       NBSEL
      LOGICAL             LOPEN
```

4.48 cafile: Subroutine cafile from library adas3xx

```
      SUBROUTINE CAFILE( IUNIT , IZ0 , IRCODE , DSNAME , DLPATH)
C
C-----------------------------------------------------------------------
C
C  ***************** FORTRAN77 SUBROUTINE: CAFILE *********************
C
C  PURPOSE: TO OPEN AN IONIZATION RATE-COEFFT 'IONELEC' DATA SET
C           BY DEFAULT, OR AN ALTERNATIVE DATA SET IF REQUIRED, FOR
C           IONIZING ION WITH NUCLEAR CHARGE 'IZ0'
C           THIS WILL BE CONNECTED TO UNIT 'IUNIT'.
C
C   DATA SET OPENED: DLPATH/adf07/<DEFADF>/<GROUP>(OPTIONAL)/<TYPE>/
C  <GROUP_EXT>#<ELEMENT SYMBOL>'
C
C  CALLING PROGRAM: CASSZD
C
C  SUBROUTINE:
C
C  INPUT : (I*4)   IUNIT   = UNIT TO WHICH DATA SET WILL BE CONNECTED
C  INPUT : (I*4)   IZ0     = NUCLEAR CHARGE OF EMITTING ION REQUESTED
C  INPUT : (C*80) DLPATH  = PATH NAME TO THE RELEVANT DATA FILES
C
C  OUTPUT: (I*4)   IRCODE  = RETURN CODE FROM SUBROUTINE:
C                            0 => DATA SET SUCCESSFULLY CONNECTED
C                            1 => REQUESTED DATA  SET  MEMBER  DOES  NOT
C                                 EXISTS - DATA SET NOT CONNECTED.
C                            9 => REQUESTED DATA  SET  EXISTS BUT CANNOT
C                                 BE OPENED.
C  OUTPUT: (C*80)  DSNAME  = NAME OF OPENED DATA SET UNDER UNIX
C
C          (I*4)   IDLEN   = LENGTH, IN BYTES, OF FIXED 'DSNAME' PREFIX
C          (I*4)   LENF1   = FIRST NON-BLANK CHR OF 'DSNAME' GROUP PART
C          (I*4)   LENF2   = LAST  NON-BLANK CHR OF 'DSNAME' GROUP PART
C          (I*4)   LENF3   = FIRST NON-BLANK CHR OF 'DSNAME' TYPE PART
C          (I*4)   LENF4   = LAST  NON-BLANK CHR OF 'DSNAME' TYPE PART
C          (I*4)   LENF5   = LAST  NON-BLANK CHR OF 'DSNAME' USERID PART
C          (I*4)   LENF6   = LAST  NON-BLANK CHR OF 'DSNAME' USERID PART
C          (I*4)   LENF7   = LAST  NON-BLANK CHR OF 'DSNAME' EXTENSION PART
C          (I*4)   LENF8   = LAST  NON-BLANK CHR OF 'DSNAME' EXTENSION PART
CA         (C*1)   HASH    = '#'   IF NON-BLANK EXT, ELSE ' '.
C          (C*2)   XFESYM  = FUNCTION - (SEE ROUTINES SECTION BELOW)
C          (C*2)   ESYM    = ELEMENT SYMBOL FOR NUCLEAR CHARGE 'IZ0'
C          (C*3)   USREXT  = ADAS SOURCE DATA FILE EXTENSION
CA         (C*80)  USERID  = ADAS SOURCE DATA USER ID
C          (C*8)   USRGRP  = ADAS SOURCE DATA GROUPNAME
CA         (C*80)  USRTYP  = ADAS SOURCE DATA TYPENAME
C    (C*5)   DEFADF  = DEFAULT DATA DIRECTORY, I.E. ADF13
C
C          (L*4)   LEXIST  = .TRUE.  => REQUESTED  DATA  SET  EXISTS.
C                            .FALSE. => REQUESTED  DATA  SET  DOES  NOT
C                                       EXIST.
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C    XXUID      ADAS      FETCHES/SETS ADAS SOURCE DATA USER ID
C          XXSSZD     ADAS      FETCHES/SETS ADAS SOURCE DATA FILENAME
C                               AND FILE EXTENSION
C          XFESYM     ADAS      CHARACTER*2 FUNCTION -
```

```
C                               GATHERS ELEMENT SYMBOL FOR NUC. CHARGE
C    XXSLEN     ADAS      FINDS FIRST AND LAST NON-BLANK
C CHARACTERS IN STRING.
C
C AUTHOR:  H.P. SUMMERS
C    K1/1/57
C    JET EXT. 4941
C DATE:    11/10/91
C
C UPDATE:  10/03/93 - PE BRIDEN: ADDED CALL TO XXUID AND USERID VARIABLE
C                                - NOW ALLOWS ANY INPUT DATASET USER ID.
C UPDATE:   2/09/93 - HPS      : ADDED CALL TO XXSSZD AND USRGRP, USRTYP
C                                  AND USREXT NAMES
C                                - NOW ALLOWS ANY INPUT DATASET FILENAME
C                                  AND EXTENSION
C UPDATE:  23/11/93 - PEB      : CORRECT ERROR - A '.' HAD MISTAKENLY
C                                BEEN PLACED BEFORE THE MEMBER NAME IN
C                                VARIABLE DSNAME.
C
C UPDATE:  10/11/94 - L. JALOTA: MODIFIED CODE FOR RUNNING UNDER UNIX
C  USING NEW FILE NAMING CONVENTION.
C  "ACTION" KEYWORD IN OPEN COMMAND IS IBM
C  SO REMOVED HERE.
C   ADDED DEFADF
C UPDATE:  22/11/94 - L. JALOTA: TIDIED UP CHARACTER LENGTH DEFINITIONS
C
C UPDATE:  24/03/95 - HPS      : INTRODUCED HASH TO ELIMINATE # IN FILE IF
C                                THERE IS NO EXTENSION PART OF THE FILE NAME
C                                ALTER LOGIC TO ALLOW USRTYP, USREXT TO BE A
C                                SINGLE CHARACTER.
C
C VERSION: 1.1                          DATE: 08-02-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST VERSION
C
C-----------------------------------------------------------------------
      CHARACTER*80        DLPATH,      DSNAME
      INTEGER             IRCODE,      IUNIT,       IZ0
```

4.49 capasf: Subroutine capasf from library adas3xx

```
      SUBROUTINE CAPASF( IUPS1  , IUPS2  , IUPS3  , IUPS4  , IURUN ,
     &                   DSNPS1 , DSNPS2 , DSNPS3 , DSNPS4 , DSNRUN,
     &                   LPASS1 , LPASS2 , LPASS3 , LPASS4 , LRUN  ,
     &                   IRCODE  )
C
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: CAPASF *********************
C
C  PURPOSE: HANDLES OPENING OF OUTPUT PASSING FILES.
C
C  CALLING PROGRAM: ADAS310
C
C  SUBROUTINE:
C
C  INPUT : (I*4)   IUPS1    = UNIT NO. FOR FIRST PASSING FILE.
C  INPUT : (I*4)   IUPS2    = UNIT NO. FOR SECOND PASSING FILE.
C  INPUT : (I*4)   IUPS3    = UNIT NO. FOR THIRD PASSING FILE.
C  INPUT : (I*4)   IUPS4    = UNIT NO. FOR FOURTH PASSING FILE.
C  INPUT : (I*4)   IURUN    = UNIT NO. FOR RUN SUMMARY FILE.
C  INPUT : (C*80)  DSNPS1   = FIRST PASSING FILE DATA SET NAME.
C  INPUT : (C*80)  DSNPS2   = SECOND PASSING FILE DATA SET NAME.
C  INPUT : (C*80)  DSNPS3   = THIRD PASSING FILE DATA SET NAME.
C  INPUT : (C*80)  DSNPS4   = FOURTH PASSING FILE DATA SET NAME.
C  INPUT : (C*80)  DSNRUN   = RUN SUMMARY FILE DATA SET NAME.
C  INPUT : (L*4)   LPASS1   = FLAG FOR WHETHER OR NOT 1ST PASSING FILE
C                             HAS BEEN REQUESTED
C  INPUT : (L*4)   LPASS2   = FLAG FOR WHETHER OR NOT 2ND PASSING FILE
C                             HAS BEEN REQUESTED
C  INPUT : (L*4)   LPASS3   = FLAG FOR WHETHER OR NOT 3RD PASSING FILE
C                             HAS BEEN REQUESTED
C  INPUT : (L*4)   LPASS4   = FLAG FOR WHETHER OR NOT 4TH PASSING FILE
C                             HAS BEEN REQUESTED
C
C  OUTPUT: (I*4)   IRCODE   = RETURN CODE AFTER ATTEMPTING TO OPEN FILES.
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C
C AUTHOR:  JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 5183
C
C DATE:    17/01/94
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 08-02-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST VERSION
C
C VERSION: 1.2                          DATE: 02-04-96
C MODIFIED: TIM HAMMOND/PAUL BRIDEN
C               - REMOVED FILE= ARGUMENT FROM SCRATCH FILE OPEN
C                 STATEMENTS AS IT CAN CAUSE A COMPILATION ERROR OF
C                 WARNING MESSAGE UNDER AIX.
C
```

```
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      CHARACTER*80        DSNPS1,      DSNPS2,      DSNPS3,      DSNPS4
      CHARACTER*80        DSNRUN
      INTEGER             IRCODE,      IUPS1,       IUPS2,       IUPS3
      INTEGER             IUPS4,       IURUN
      LOGICAL             LPASS1,      LPASS2,      LPASS3,      LPASS4
      LOGICAL             LRUN
```

4.50 casszd: Subroutine casszd from library adas3xx

```
      SUBROUTINE CASSZD( IBSEL  , IZ0IN  ,
     &                   ITVAL  , TVAL   ,
     &                   BWNO   , IZ     , IZ1  ,
     &                   METI   , METF   ,
     &                   SZDA   , LTRNG  ,
     &                   TITLX  , IRCODE , DLPATH
     &                 )

C-----------------------------------------------------------------------
C
C  ******************* FORTRAN77 SUBROUTINE: CASSZD *******************
C
C  PURPOSE: TO EXTRACT AND  INTERPOLATE  ZERO-DENSITY  IONIZATION  RATE-
C           COEFFICIENTS FOR GIVEN ELEMENT NUCLEAR CHARGE AND DATA-BLOCK
C           FOR AN INPUT SET OF ELECTRON TEMPERATURES (eV).
C
C           - USES THE SAME ROUTINES USED BY SSZD, EXCEPT FOR:
C
C           'CAFILE' - WHICH OPENS THE REQUESTED FILE.
C           'CACHKB' - WHICH CHECKS INPUT  VALUES  ARE  CONSISTENT  WITH
C                      THE SELECTED DATA-BLOCK 'IBSEL' AND   'IBSEL'  IS
C                      IN RANGE.
C
C            THE FIRST OF THESE FUNCTIONS IS CARRIED  OUT  IN  'ADAS502'
C            VIA ISPF PANELS USING THE ROUTINE 'E2SPF0'  -  ADAS502 DOES
C            NOT REQUIRE THE ROUTINE 'CACHKB' AS THE USER CANNOT  SELECT
C            AN INVALID VALUE FOR 'IBSEL' OR 'IBSEL'/ELEMENT COMBINATION
C
C
C  CALLING PROGRAM: NSUPH1
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  IBSEL   = INDEX OF DATA-BLOCK SELECTED FOR ANALYSIS
C  INPUT : (I*4)  IZ0IN   = NUCLEAR CHARGE OF REQUIRED ELEMENT
C
C  INPUT : (I*4)  ITVAL   = NUMBER OF ELECTRON TEMPERATURE VALUES
C  INPUT : (R*8)  TVAL()  = ELECTRON TEMPERATUIRES (UNITS: EV)
C                           DIMENSION: ELECTRON TEMPERATURE INDEX
C  INPUT : (C*80) DLPATH  = PATH NAME TO THE RELEVANT DATA FILES
C                           (PASSED THROUGH TO CAFILE TO BUILD FILENAME)
C
C  OUTPUT: (R*8)  BWNO    = INPUT FILE - SELECTED DATA-BLOCK:
C                           EFFECTIVE IONIZATION POTENTIAL (cm-1).
C  OUTPUT: (I*4)  IZ      = INPUT FILE - SELECTED DATA BLOCK:
C                           IONIZING ION - INITIAL CHARGE
C  OUTPUT: (I*4)  IZ1     = INPUT FILE - SELECTED DATA BLOCK:
C                           IONIZING ION - FINAL   CHARGE
C
C  OUTPUT: (I*4)  METI    = INPUT FILE - SELECTED DATA-BLOCK:
C                           INITIAL STATE METSTABLE INDEX
C  OUTPUT: (I*4)  METF    = INPUT FILE - SELECTED DATA-BLOCK:
C                           FINAL   STATE METSTABLE INDEX
C
C  OUTPUT: (R*8)  SZDA()  = ZERO-DENSITY IONIZATION RATE-COEFFICIENTS
C                           DIMENSION: ELECTRON TEMPERATURE INDEX
C  OUTPUT: (L*4)  LTRNG() =.TRUE.  => OUTPUT 'SZDA()'  VALUE WAS INTER-
C                                     POLATED  FOR  THE  USER  ENTERED
C                                     ELECTRON TEMPERATURE 'TVAL()'.
```

```
C                            .FALSE. => OUTPUT 'SZDA()'  VALUE WAS EXTRA-
C                                       POLATED  FOR  THE  USER  ENTERED
C                                       ELECTRON TEMPERATURE 'TVAL()'.
C                             DIMENSION: ELECTRON TEMPERATURE INDEX
C
C  OUTPUT: (C*80) TITLX   = INFORMATION STRING (DSN ETC.)
C  OUTPUT: (I*4)  IRCODE  = RETURN CODE FROM SUBROUTINE:
C                             0 => NORMAL COMPLETION - NO ERROR DETECTED
C                             1 => DATA SET MEMBER FOR IONIZING ION WITH
C                                  NUCLEAR CHARGE  'IZ0IN'  CAN  NOT  BE
C                                  FOUND/DOES NOT EXIST.
C                             2 => DISCREPANCY BETWEEN REQUESTED CHARGES
C                                  AND THOSE IN INPUT FILE.
C                             3 => THE SELECTED DATA-BLOCK 'IBSEL' IS OUT
C                                  OF RANGE OR DOES NOT EXIST.
C                             4 => INVALID VALUE FOR 'IZ0IN' ENTERED.
C                                  ('IZ0MIN' <= 'IZ0IN' <= 'IZ0MAX')
C                             9 => ERROR ENCOUNTERED WHEN TRYING TO OPEN
C                                  INPUT DATA-SET.
C
C          (I*4)  NSTORE  = PARAMETER= MAXIMUM NUMBER  OF  DATA-BLOCKS
C                                      WHICH CAN BE READ FROM THE INPUT
C                                      DATA-SET.
C          (I*4)  NTDIM   = PARAMETER= MAXIMUM NUMBER OF ELECTRON TEMP-
C                                      ERATURES THAT CAN BE READ  FROM
C                                      AN INPUT DATA-SET DATA-BLOCK.
C          (I*4)  IZ0MIN  = PARAMETER: MIN. ALLOWED VALUE FOR 'IZ0IN'
C          (I*4)  IZ0MAX  = PARAMETER: MAX. ALLOWED VALUE FOR 'IZ0IN'
C
C          (I*4)  IZ0LST  = LAST VALUE OF 'IZ0IN' FOR  WHICH  INPUT
C                           DATA WAS READ.
C          (I*4)  IUNIT   = UNIT TO WHICH INPUT DATA SET IS ALLOCATED
C          (I*4)  NBSEL   = TOTAL NUMBER OF DATA-BLOCKS READ FROM INPUT
C                           DATA SET.
C          (I*4)  IZ0     = INPUT FILE - EMITTING ION - NUCLEAR CHARGE
C
C          (L*4)  LOPEN   = .TRUE.  => INPUT DATA SET OPEN.
C                           .FALSE. => INPUT DATA SET CLOSED.
C
C          (C*2)  ESYM    = INPUT FILE - IONIZING ION - ELEMENT SYMBOL
C          (C*3)  EXTIN   = CURRENT ADAS SOURCE DATA FILE EXTENSION
C          (C*3)  EXTLST  = ADAS SOURCE DATA FILE EXT. USED LAST TIME
C                           DATA WAS READ.
C          (C*6)  UIDIN   = CURRENT ADAS SOURCE DATA USER ID.
CA         (C*80) UIDIN   = CURRENT ADAS SOURCE DATA FILE PATH
CA         (C*80) UIDLST  = ADAS SOURCE DATA FILE PATH USED LAST TIME
C                           DATA WAS READ.
C          (C*8)  GRPIN   = CURRENT ADAS SOURCE DATA GROUPNAME
C          (C*8)  GRPLST  = ADAS SOURCE DATA GROUPNAME USED LAST TIME
C                           DATA WAS READ.
CA         (C*80) TYPIN   = CURRENT ADAS FILE SUBDIRECTORY( OPTIONAL)
CA         (C*80) TYPLST  = ADAS FILE SUBDIRECTORY USED LAST TIME (OPT)
C                           DATA WAS READ.
CA    (C*80) DSNREQ  = DATAFILE NAME UNDER UNIX INCLUDING PATH
C                           (MAY OR MAY NOT EXIST)
CA         (C*80) DSNAME  = DATAFILE NAME UNDER UNIX INCLUDING PATH
C
C          (I*4)  ISELA() = INPUT DATA FILE: DATA-BLOCK ENTRY INDICES.
C                           DIMENSION: DATA-BLOCK INDEX
C          (I*4)  ITA()   = INPUT DATA SET-NUMBER OF ELECTRON TEMPERA-
C                           TURES.
```

```
C                           DIMENSION: DATA-BLOCK INDEX
C          (I*4)  IZOUT() = INPUT DATA FILE: IONIZING ION INITIAL CHARGE
C                           DIMENSION: DATA-BLOCK INDEX
C          (I*4)  IZ1OUT()= INPUT DATA FILE: IONIZING ION FINAL   CHARGE
C                           DIMENSION: DATA-BLOCK INDEX
C
C          (R*8)  BWNOUT()= INPUT DATA FILE: EFFECTIVE IONIZATION POT.
C                           (UNITS: cm-1).
C                           DIMENSION: DATA-BLOCK INDEX
C          (R*8)  TETA(,) = INPUT DATA SET -
C                           ELECTRON TEMPERATURES (UNITS: eV)
C                           1st DIMENSION: ELECTRON TEMPERATURE INDEX
C                           2nd DIMENSION: DATA-BLOCK INDEX
C          (R*8)  SZD(,)    =INPUT DATA SET -
C                            FULL SET OF IONIZATIONS RATE-COEFFICIENTS
C                            1st DIMENSION: ELECTRON TEMPERATURE INDEX
C                            3rd DIMENSION: DATA-BLOCK INDEX
C
C          (C*2)  CICODE()= INPUT DATA FILE - INITIAL STATE META. INDEX
C                           DIMENSION: DATA-BLOCK INDEX
C          (C*2)  CFCODE()= INPUT DATA FILE - FINAL   STATE META. INDEX
C                           DIMENSION: DATA-BLOCK INDEX
C          (C*6)  CIION() = INPUT DATA FILE - INITIAL ION
C                           DIMENSION: DATA-BLOCK INDEX
C          (C*6)  CFION() = INPUT DATA FILE - FINAL   ION
C                           DIMENSION: DATA-BLOCK INDEX
C
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          CAFILE     ADAS      OPEN DATA SET FOR SELECTED ELEMENT
C          XXDATA_07  ADAS      FETCH INPUT DATA FROM SELECTED DATA SET
C          CACHKB     ADAS      CHECK VALIDITY OF ELEMENT AND 'IBSEL'
C          E2SPLN     ADAS      INTERPOLATE DATA WITH ONE-WAY SPLINES
C          E2TITL     ADAS      CREATE DESCRIPTIVE TITLE FOR OUTPUT
C          XXUID      ADAS      FETCHES/SETS ADAS SOURCE DATA USER ID
C          XXSPEC     ADAS      FETCHES/SETS ADAS SOURCE DATA FILE NAME+
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/37
C          JET EXT. 6023
C
C DATE:    07/06/91
C
C UPDATE:  06/12/91 - PE BRIDEN: 'NSTORE' INCREASED FROM 10 TO 100
C
C UPDATE:  28/02/92 - PE BRIDEN: 'NSTORE' INCREASED FROM 100 TO 160
C
C UPDATE:  10/03/93 - PE BRIDEN: INTRODUCED CALL TO XXUID TO ESTABLISH
C                                IF USERID OF INPUT DATASET CHANGES
C                                BETWEEN CALLS.
C                                SAVE NAME OF LAST READ DATASET.
C                                (ADDED VARIABLES UIDIN,UIDLST,DSNREQ)
C
C UPDATE:   2/09/93 - HPS      : INTRODUCED CALL TO XXSSZD TO ESTABLISH
C                                IF USRGRP, USRTYP AND USREXT OF INPUT
C                                DATASET CHANGES BETWEEN CALLS.
C                                SAVE NAME OF LAST READ DATASET.
C                                (ADDED VARIABLES GRPIN,GRPLST,TYPIN,
C                                 TYPLST, EXTIN, EXTLST)
```

```
C
C UPDATE:   10/11/94 - L. JALOTA: MODIFIED TO RUN UNDER UNIX, SIZE OF
C   DSNAME AND DSNREQ INCREASED TO 80
C   CHARACTERS
C
C UPDATE:   21/11/94 - L/ JALOTA: TIDIED UP CHARACTER LENGTHS.
C
C UNIX-IDL PORT:
C
C
C VERSION: 1.1                          DATE: 25-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - CREATED FROM SSZD.FOR WITH SPECIFIC USE IN ADAS 310
C                 AS GOAL. ALL FUNCTIONALITY IS MAINTAINED, BUT
C                 ROUTINE CALLS A NEW FILE CAFILE.FOR TO OPEN
C                 THE REQUESTED FILE IF IT IS AVAILABLE, THIS
C                 HAS ALSO NECESSITATED BRINGING IN THE VARIABLE
C                 DLPATH WHICH HOLDS THE PATH TO THE DATA FILES.
C
C
C VERSION : 1.2
C DATE    : 20-10-2003
C MODIFIED: Martin O'Mullane
C           - Extend TITLX to 120 to match e2titl routine.
C           - Save essential variable between calls.
C
C VERSION : 1.3
C DATE    : 17-05-2007
C MODIFIED: Allan Whiteford
C           - Updated comments as part of subroutine documentation
C             procedure.
C
C VERSION : 1.4
C DATE    : 26-03-2008
C MODIFIED: Allan Whiteford
C           - Changed call from E2DATA to XXDATA_07
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      CHARACTER*80        DLPATH
      CHARACTER*120       TITLX
      INTEGER             IBSEL,       IRCODE,      ITVAL,       IZ
      INTEGER             IZ0IN,       IZ1,         METF,        METI
      LOGICAL             LTRNG(ITVAL)
      REAL*8              BWNO,        SZDA(ITVAL), TVAL(ITVAL)
```

4.51 catmpf: Subroutine catmpf from library adas3xx

```
      SUBROUTINE CATMPF( IUTMP  , NUCCHG , DSNEX  , DSNCX  ,
     &                   JDENSM , JTEM   , TS     , W      ,
     &                   Z      , CION   , CPY    , W1     ,
     &                   NIP    , INTD   , IPRS   , ILOW   ,
     &                   IONIP  , NIONIP , ILPRS  , IVDISP ,
     &                   ZEFF   , NOSCAN , NIMP   , ZIMPA  ,
     &                   AMIMPA , FRIMPA , DENSA  , TEA    ,
     &                   DENPA  , TPA    , BMENER , DENSH  ,
     &                   NMIN   , NMAX   , IMAX   , NREP   ,
     &                   WBREP  , JCOR   , COR    , JMAX   ,
     &                   EPSIL  , FIJ    , WIJ    , JDEF   ,
     &                   DEFECT
     &                 )
C
C
C-----------------------------------------------------------------------
C
C ******************* FORTRAN77 SUBROUTINE: CATMPF *********************
C
C PURPOSE: TO WRITE TEMPORARY FILE CONTAINING PARAMETERS TO BE READ BY
C          SUBROUTINE 'V2BNDLN'.
C
C          THIS ROUTINE IS A TEMPORARY MEASURE TO ALLOW 'V2BNDLN' TO
C          RUN WITHIN THE FRAMEWORK OF ADAS WITHOUT HAVING TO EDIT
C          'V2BNDLN'. THE PARAMETERS SHOULD REALLY BE PASSED INTO
C          'V2BNDLN' THROUGH ITS ARGUMENT LIST.
C
C CALLING PROGRAM: ADAS310
C
C  INPUT : (I*4)   IUTMP    = UNIT NUMBER OF TEMPORARY FILE.
C  INPUT : (I*4)   NUCCHG   = NUCLEAR CHARGE.
C  INPUT : (C*80)  DSNEX    = FULL MVS DATA SET NAME FOR EXPANSION FILE
C                             (SUITABLE FOR DYNAMIC ALLOCATION).
C  INPUT : (C*80)  DSNCX    = FULL MVS DATA SET NAME FOR CHARGE EXCHANGE
C                             DATA SET (SUITABLE FOR DYNAMIC ALLOCATION)
C  INPUT : (I*4)   JDENSM   = NUMBER OF DENSITIES.
C  INPUT : (I*4)   JTEM     = NUMBER OF TEMPERATURES.
C  INPUT : (R*8)   TS       = EXTERNAL RADIATION FIELD TEMPERATURE.
C                             UNITS: K
C  INPUT : (R*8)   W        = EXTERNAL RADIATION FIELD DILUTION FACTOR
C                             (HIGHER LEVELS).
C  INPUT : (R*8)   Z        = RECOMBINING ION CHARGE.
C  INPUT : (R*8)   CION     = MULTIPLIER OF GROUND LEVEL ELECTRON IMPACT
C                             IONISATION RATE COEFFICIENT.
C  INPUT : (R*8)   CPY      = MULTIPLIER OF ELECTRON EXCITATION RATE
C                             COEFFICIENT FROM THE GOUND LEVEL.
C  INPUT : (R*8)   W1       = EXTERNAL RADIATION FIELD DILUTION FACTOR
C                             FOR PHOTO-IONISATION FROM THE GROUND
C                             LEVEL.
C  INPUT : (R*8)   NIP      = RANGE OF DELTA N FOR IMPACT PARAMETER
C                             XSECTS. (.LE. 4)
C  INPUT : (R*8)   INTD     = ORDER OF MAXWELL QUADRATURE FOR XSECTS.
C                             (.LE. 3)
C  INPUT : (R*8)   IPRS     = CONTROLS XSECTS BEYOND NIP RANGE.
C                             0 => DEFAULT TO VAN REGEMORTER XSECTS.
C                             1 => USE PERCIVAL-RICHARDS XSECTS.
C  INPUT : (R*8)   ILOW     = CONTROLS ACCESS OF SPECIAL LOW LEVEL DATA.
C                             0 => NO SPECIAL LOW LEVEL DATA ACCESSED.
C                             1 => SPECIAL LOW LEVEL DATA ACCESSED.
```

```
C  INPUT : (R*8)   IONIP    = CONTROLS INCLUSION OF ION IMPACT
C                             COLLISIONS.
C                             0 => NO ION IMPACT COLLISIONS INCLUDED.
C                             1 => ION IMPACT EXCITATION AND IONISATION
C                                  INCLUDED.
C  INPUT : (R*8)   NIONIP   = RANGE OF DELTA N FOR ION IMPACT EXCITATION
C                             XSECTS.
C  INPUT : (R*8)   ILPRS    = CONTROLS USE OF LODGE-PERCIVAL-RICHARDS
C                             XSECTS.
C                             0 => DEFAULT TO VAINSHTEIN XSECTS.
C                             1 => USE LODGE-PERCIVAL-RICHARDS XSECTS.
C  INPUT : (R*8)   IVDISP   = CONTROLS USE OF BEAM ENERGY IN CALCULATION
C                             OF XSECTS.
C                             0 => ION IMPACT AT THERMAL MAXWELLIAN
C                                  ENERGIES.
C                             1 => ION IMPACT AT DISPLACED THERMAL
C                                  ENERGIES ACCORDING TO THE NEUTRAL
C                                  BEAM ENERGY PARAMETER.
C                             NB:  IF IVDISP=0 THEN SPECIAL LOW LEVEL
C                                  DATA FOR ION IMPACT IS NOT
C                                  SUBSTITUTED - ONLY VAINSHTEIN AND
C                                  LODGE ET AL. OPTIONS ARE OPEN.
C                                  ELECTRON IMPACT DATA SUBSTITUTION
C                                  DOES OCCUR.
C  INPUT : (R*8)   ZEFF     = NUCLEAR CHARGE OF IMPURITY.
C                             (ONLY SET IF 'NOSCAN'=0 )
C  INPUT : (I*4)   NOSCAN   = CONTROLS MODE OF OPERATION.
C                             0 => SINGLE IMPURITY.
C                             1 => MULTIPLE IMPURITIES.
C  INPUT : (I*4)   NIMP     = NUMBER OF IMPURITY SPECIES
C                             (ONLY SET IF 'NOSCAN'=1 )
C  INPUT : (R*8)   ZIMPA()  = NUCLEAR CHARGE OF IMPURITIES.
C                             (ONLY SET IF 'NOSCAN'=1 )
C                             DIMENSION: NIMP
C  INPUT : (R*8)   AMIMPA() = ATOMIC MASS NUMBERS OF IMPURITIES.
C                             (ONLY SET IF 'NOSCAN'=1 )
C                             DIMENSION: NIMP
C  INPUT : (R*8)   FRIMPA() = IMPURITY FRACTIONS.
C                             (ONLY SET IF 'NOSCAN'=1 )
C                             DIMENSION: NIMP
C  INPUT : (R*8)   DENSA()  = ELECTRON DENSITIES.
C                             UNITS: CM-3
C                             DIMENSION: JDENSM
C  INPUT : (R*8)   TEA()    = ELECTRON TEMPERATURES.
C                             UNITS: K
C                             DIMENSION: JTEM
C  INPUT : (R*8)   DENPA()  = PROTON DENSITIES.
C                             UNITS: CM-3
C                             DIMENSION: JDENSM
C  INPUT : (R*8)   TPA()    = PROTON TEMPERATURES.
C                             UNITS: K
C                             DIMENSION: JTEM
C  INPUT : (R*8)   BMENER   = NEUTRAL BEAM PARTICLE ENERGY.
C                             UNITS: EV / AMU
C  INPUT : (R*8)   DENSH    = NEUTRAL HYDROGEN DENSITY IN BEAM.
C                             UNITS: CM-3
C  INPUT : (I*4)   NMIN     = LOWEST N-SHELL.
C  INPUT : (I*4)   NMAX     = HIGHEST N-SHELL.
C  INPUT : (I*4)   IMAX     = NUMBER OF REPRESENTATIVE N-SHELL LEVELS.
C  INPUT : (I*4)   NREP()   = SET OF REPRESENTATIVE N-SHELL LEVELS.
C                             DIMENSION: IMAX
```

```
C  INPUT : (R*8)   WBREP()  =
C                             DIMENSION: IMAX
C  INPUT : (I*4)   JCOR     =
C  INPUT : (R*8)   COR()    =
C                             DIMENSION: JCOR
C  INPUT : (I*4)   JMAX     =
C  INPUT : (R*8)   EPSIL()  =
C                             DIMENSION: JMAX
C  INPUT : (R*8)   FIJ()    =
C                             DIMENSION: JMAX
C  INPUT : (R*8)   WIJ()    =
C                             DIMENSION: JMAX
C  INPUT : (I*4)   JDEF     = NUMBER OF QUAMTUM DEFECTS.
C  INPUT : (R*8)   DEFECT() = SET OF QUANTUM DEFECT.
C                             DIMENSION: JDEF
C
C          (I*4)   I        = ARRAY INDEX.
C
C          (L*4)   LOPEN    = FLAGS IF SCRATCH FILE OPEN.
C                             .TRUE.  => SCRATCH FILE OPEN.
C                             .FALSE. => SCRATCH FILE CLOSED.
C
C AUTHOR:  JONATHAN NASH  (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 5183
C
C DATE:    17/01/94
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 08-02-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST VERSION
C
C VERSION: 1.2                          DATE: 17-05-07
C MODIFIED: Allan Whiteford
C               - Updated comments as part of subroutine documentation
C                 procedure.
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      CHARACTER*80        DSNCX,       DSNEX
      INTEGER             ILOW,        ILPRS,       IMAX,        INTD
      INTEGER             IONIP,       IPRS,        IUTMP,       IVDISP
      INTEGER             JCOR,        JDEF,        JDENSM,      JMAX
      INTEGER             JTEM,        NIMP,        NIONIP,      NIP
      INTEGER             NMAX,        NMIN,        NOSCAN
      INTEGER             NREP(IMAX),  NUCCHG
      REAL*8              AMIMPA(NIMP),             BMENER,      CION
      REAL*8              COR(JCOR),   CPY,         DEFECT(JDEF)
      REAL*8              DENPA(JDENSM),            DENSA(JDENSM)
      REAL*8              DENSH,       EPSIL(JMAX), FIJ(JMAX)
      REAL*8              FRIMPA(NIMP),             TEA(JTEM)
      REAL*8              TPA(JTEM),   TS,          W,           W1
      REAL*8              WBREP(IMAX), WIJ(JMAX),   Z,           ZEFF
      REAL*8              ZIMPA(NIMP)
```

4.52 cbpasf: Subroutine cbpasf from library adas3xx

```
      SUBROUTINE CBPASF( IUPS1  , IUPS2  , IURUN , DSNPS1 , DSNPS2 ,
     &                   DSNRUN , LPASS1 , LPASS2 , LRUN  , IRCODE  )
C
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: CBPASF *********************
C
C  PURPOSE: HANDLES OPENING OF OUTPUT PASSING FILES.
C
C  CALLING PROGRAM: ADAS311
C
C  SUBROUTINE:
C
C  INPUT : (I*4)   IUPS1    = UNIT NO. FOR FIRST PASSING FILE.
C  INPUT : (I*4)   IUPS2    = UNIT NO. FOR SECOND PASSING FILE.
C  INPUT : (I*4)   IURUN    = UNIT NO. FOR RUN SUMMARY FILE.
C  INPUT : (C*80)  DSNPS1   = FIRST PASSING FILE DATA SET NAME.
C  INPUT : (C*80)  DSNPS2   = SECOND PASSING FILE DATA SET NAME.
C  INPUT : (C*80)  DSNRUN   = RUN SUMMARY FILE DATA SET NAME.
C  INPUT : (L*4)   LPASS1   = FLAG FOR WHETHER OR NOT 1ST PASSING FILE
C                             HAS BEEN REQUESTED
C  INPUT : (L*4)   LPASS2   = FLAG FOR WHETHER OR NOT 2ND PASSING FILE
C                             HAS BEEN REQUESTED
C
C  OUTPUT: (I*4)   IRCODE   = RETURN CODE AFTER ATTEMPTING TO OPEN FILES.
C
C
C  CONTACT : HARVEY ANDERSON
C      UNIVERSITY OF STRATHCLYDE
C      ANDERSON@PHYS.STRATH.AC.UK
C
C  NOTE   : ROUTINE BASED ON THE STRUCTURE OF CAPASF.F, WHICH IS
C           USED BY ADAS310.
C
C  DATE   :  02/15/94
C
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      CHARACTER*80        DSNPS1,      DSNPS2,      DSNRUN
      INTEGER             IRCODE,      IUPS1,       IUPS2,       IURUN
      LOGICAL             LPASS1,      LPASS2,      LRUN
```

4.53 ccaval: Subroutine ccaval from library adas3xx

```
        SUBROUTINE CCAVAL(NUPPER,NLOWER,AVALUE)
C-----------------------------------------------------------------------
C
C   *********   FORTRAN77 SUBROUTINE: CCAVAL *********
C
C   PURPOSE:  EVALUATES HYDROGENIC TRANISITON PROBABILITIES
C   FOR TRANSITIONS BETWEEN PRINCIPAL QUANTUM SHELLS.
C
C   CALLING PROGRAM: ADAS312
C
C INPUT  :
C
C (I*4) NUPPER : UPPER PRINCIPAL QUANTUM
C   NUMBER.
C (I*4) NLOWER : LOWER PRINCIPAL QUANTUM
C   NUMBER.
C
C OUPUT :
C (R*8) AVALUE : TRANSITION PROBABILITY.
C
C GENERAL :
C
C (R*8) EN : UPPER PRINCIPAL QUANTUM
C   NUMBER.
C (R*8) EN1 : LOWER PRINCIPAL QUANTUM
C   NUMBER.
C (R*8) X2 : GENERAL VARIABLE
C (R*8) X : GENERAL VARIABLE
C (R*8)   G : BOUND-BOUND GAUNT FACTOR.
C
C
C  ROUTINES:
C
C          ROUTINE         SOURCE       BRIEF DESCRIPTION
C          -------         ------       -----------------
C     GBB     ADAS EVALUATES BOUND-BOUND
C GAUNT FACTOR.
C
C  CONTACT : HARVEY ANDERSON
C        UNIVERSITY OF STRATHCLYDE
C           ANDERSON@PHYS.STRATH.AC.UK
C
C  DATE    : 11/03/99
C
C VERSION: 1.1 DATE: 17-03-99
C MODIFIED: HARVEY ANDERSON
C - FIRST VERSION
C
C-----------------------------------------------------------------------

      INTEGER             NLOWER,      NUPPER
      REAL*8              AVALUE
```

4.54 ccdata: Subroutine ccdata from library adas3xx

```
       SUBROUTINE CCDATA( TERAY  , NERAY  , EBRAY  , N1N    , SRAY  ,
     &                    F1     , F2     , F3     , BN     ,
     &   NN     , IZ     , INFILE , INUNIT ,
     &                    MAXNE  , MAXTE  , MAXEB  ,
     &   INCOUNT, ITCOUNT, IECOUNT, LEVEL  ,
     &                    EBREF  , TEREF  , NEREF  ,
     &                    INA    , IEA    , ITA    , NLEVEL , FLAG  )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: CCDATA *********************
C
C  PURPOSE:  TO  FETCH  DATA  FROM  BUNDLE-N POPULATION
C            FILES OF TYPE ADF26.
C
C  CALLING PROGRAM: ADAS312
C
C  SUBROUTINE:
C
C  INPUT : (C*80) INFILE   = MVS DATA SET NAME OF DATA SET BEING READ
C  INPUT : (I*4)  INUNIT   = UNIT TO WHICH INPUT FILE IS ALLOCATED.
C  INPUT : (I*4)  MAXNE    = MAXIMUM NUMBER OF DENSITIES
C  INPUT : (I*4)  MAXTE    = MAXIMUM NUMBER OF TEMPERATURES
C  INPUT : (I*4)  MAXEB    = MAXIMUM NUMBER OF BEAM ENERGIES
C
C  OUTPUT: (R*8)  TERAY()  = TEMPERATURE SET   (EV)
C                            1ST. DIM.: TEMPERATURE INDEX
C  OUTPUT: (R*8)  NERAY()  = DENSITY SET       (CM-3)
C                            1ST. DIM.: DENSITY INDEX
C  OUTPUT: (R*8)  EBRAY()  = BEAM ENERGY SET   (EV/AMU)
C                            1ST. DIM.: BEAM ENERGY INDEX
C  OUTPUT: (R*8)  N1N(,,)  = ???
C                            1ST. DIM.: BEAM ENERGY INDEX
C                            2ND. DIM.: DENSITY INDEX
C                            3RD. DIM.: TEMPERATURE INDEX
C  OUTPUT: (R*8)  SRAY(,,) = COLL. RAD. IONIS COEFFT.   (CM3 S-1)
C                            1ST. DIM.: BEAM ENERGY INDEX
C                            2ND. DIM.: DENSITY INDEX
C                            3RD. DIM.: TEMPERATURE INDEX
C  OUTPUT: (R*8)  F1(,,,)  = F1 EXPANSION FACTOR OF BN
C                            1ST. DIM.: BEAM ENERGY INDEX
C                            2ND. DIM.: DENSITY INDEX
C                            3RD. DIM.: TEMPERATURE INDEX
C                            4TH. DIM.: REPRES. PRINC. QUANTUM SHELL INDEX
C  OUTPUT: (R*8)  F2(,,,)  = F2 EXPANSION FACTOR OF BN
C                            1ST. DIM.: BEAM ENERGY INDEX
C                            2ND. DIM.: DENSITY INDEX
C                            3RD. DIM.: TEMPERATURE INDEX
C                            4TH. DIM.: REPRES. PRINC. QUANTUM SHELL INDEX
C  OUTPUT: (R*8)  F3(,,,)  = F3 EXPANSION FACTOR OF BN
C                            1ST. DIM.: BEAM ENERGY INDEX
C                            2ND. DIM.: DENSITY INDEX
C                            3RD. DIM.: TEMPERATURE INDEX
C                            4TH. DIM.: REPRES. PRINC. QUANTUM SHELL INDEX
C  OUTPUT: (R*8)  BN(,,,)  = BN FACTOR
C                            1ST. DIM.: BEAM ENERGY INDEX
C                            2ND. DIM.: DENSITY INDEX
C                            3RD. DIM.: TEMPERATURE INDEX
C                            4TH. DIM.: REPRES. PRINC. QUANTUM SHELL INDEX
C  OUTPUT: (R*8)  NN(,,,)  = POPULATION CONVERSION FACTOR
```

```
C                             1ST. DIM.: BEAM ENERGY INDEX
C                             2ND. DIM.: DENSITY INDEX
C                             3RD. DIM.: TEMPERATURE INDEX
C                             4TH. DIM.: REPRES. PRINC. QUANTUM SHELL INDEX
C  OUTPUT: (I*4)  IZ       = IMPURITY ION CHARGE
C  OUTPUT: (I*4)  INCOUNT  = NUMBER OF DENSITIES
C  OUTPUT: (I*4)  ITCOUNT  = NUMBER OF TEMPERATURES
C  OUTPUT: (I*4)  IECOUNT  = NUMBER OF BEAM ENERGIES
C  OUTPUT: (I*4)  LEVEL    = ??? APPEARS UNUSED ???
C  OUTPUT: (R*8)  EBREF    = REFERENCE BEAM ENERGY (EV/AMU)
C  OUTPUT: (R*8)  TEREF    = REFERENCE TEMPERATURE (EV)
C  OUTPUT: (R*8)  NEREF    = REFERENCE DENSITY     (CM-3)
C  OUTPUT: (I*4)  INA()    = NUMBER OF LEVELS
C                             1ST. DIM.: DENSITY INDEX
C  OUTPUT: (I*4)  ITA()    = NUMBER OF LEVELS
C                             1ST. DIM.: TEMPERATURE INDEX
C  OUTPUT: (I*4)  IEA()    = NUMBER OF LEVELS
C                             1ST. DIM.: BEAM ENERGY INDEX
C  OUTPUT: (I*4)  NLEVEL   = NUMBER OF LEVELS
C
C  OUTPUT: (I*4)  FLAG()   = ARRAY INDICTAING WHETHER DATA HAS BEEN
C      EXTRACTED FROM THE ADF26 TYPE FILE.
C
C          (I*4)  IT       = GENERAL INDEX
C          (I*4)  IN       = GENERAL INDEX
C          (I*4)  IE       = GENERAL INDEX
C          (R*8)  TE       = GENERAL REAL VARIABLE
C          (R*8)  NE       = GENERAL REAL VARIABLE
C          (R*8)  EB       = GENERAL REAL VARIABLE
C          (C*132)LINE     = GENERAL STRING
C          (C*2)  LEVELS() = PRINC. QU. SHELL STRINGS
C
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4UNIT     ADAS      FETCH UNIT NUMBER FOR OUTPUT OF MESSAGES
C          CCFIND     ADAS      ???
C
C
C
C AUTHOR:  HARVEY ANDERSON, UNIVERSITY OF STRATHCLYDE/JET
C          JA8.08
C          TEL. 0141-553-4196
C
C
C DATE:    16/05/97
C
C UNIX-IDL PORT: H.P.SUMMERS
C
C VERSION: 1.1 DATE: 10-07-97
C MODIFIED: HUGH SUMMERS, UNIVERSITY OF STRATHCLYDE
C     - PUT UNDER S.C.C.S. CONTROL
C
C VERSION: 1.2 DATE: 15-03-99
C MODIFIED: HARVEY ANDERSON, UNIVERSITY OF STRATHCLYDE
C     - INTRODUCED THE ARRAY FLAG, WHICH IS EMPLOYED TO
C       INDICATE THE DATA WHICH HAS BEEN EXTRACTED FROM
C       THE ADF26 TYPE FILE.
C
C VERSION: 1.3 DATE: 28-04-99
```

```
C MODIFIED: RICHARD MARTIN
C   ADDED MISSING CARRIAGE RETURN AFTER FORMAT STATEMENT 1006
C
C-----------------------------------------------------------------------
      CHARACTER*80        INFILE
      INTEGER             FLAG(NLEVEL),              IEA(MAXEB),  IECOUNT
      INTEGER             INA(MAXNE),  INCOUNT,      INUNIT
      INTEGER             ITA(MAXTE),  ITCOUNT,      IZ,          LEVEL
      INTEGER             MAXEB,       MAXNE,        MAXTE,       NLEVEL
      REAL*8              BN(MAXEB,MAXNE,MAXTE,NLEVEL)
      REAL*8              EBRAY(MAXEB),              EBREF
      REAL*8              F1(MAXEB,MAXNE,MAXTE,NLEVEL)
      REAL*8              F2(MAXEB,MAXNE,MAXTE,NLEVEL)
      REAL*8              F3(MAXEB,MAXNE,MAXTE,NLEVEL)
      REAL*8              N1N(MAXEB,MAXNE,MAXTE),    NERAY(MAXNE)
      REAL*8              NEREF,       NN(MAXEB,MAXNE,MAXTE,NLEVEL)
      REAL*8              SRAY(MAXEB,MAXNE,MAXTE),   TERAY(MAXTE)
      REAL*8              TEREF
```

4.55 ccfill: Subroutine ccfill from library adas3xx

```
      SUBROUTINE CCFILL( NTA   , M )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: CCFILL *********************
C
C  PURPOSE:  TO FILL AN INTEGER ARRAY WITH ITS OWN INDEX.
C
C  CALLING PROGRAM: ADAS312
C
C  SUBROUTINE:
C
C  I/O   : (I*4)  NTA()    = ARRAY TO BE FILLED
C  INPUT : (I*4)  M        = NUMBER OF ELEMENTS TO BE FILLED
C
C          (I*4)  I        = GENERAL INTEGER
C
C
C ROUTINES:
C          NONE
C
C
C AUTHOR:  HARVEY ANDERSON, UNIVERSITY OF STRATHCLYDE/JET
C          JA8.08
C          TEL. 0141-553-4196
C
C
C DATE:    16/05/97
C
C UNIX-IDL PORT: H.P.SUMMERS
C
C VERSION: 1.1 DATE: 10-07-97
C MODIFIED: HUGH SUMMERS, UNIVERSITY OF STRATHCLYDE
C     - PUT UNDER S.C.C.S. CONTROL
C
C-----------------------------------------------------------------------
      INTEGER             M,           NTA(*)
```

4.56 ccfind: Subroutine ccfind from library adas3xx

```
      SUBROUTINE CCFIND( ARR    , VALUE  , IMAX   , INDEX  )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: CCFIND *********************
C
C  PURPOSE:  TO  ISOLATE DATA FROM ADF26 DATASET
C
C  CALLING PROGRAM: CCDATA
C
C  SUBROUTINE:
C
C  INPUT : (R*8)  ARR()    = ARRAY TO BE SEARCHED
C                            1ST. DIM.:
C  INPUT : (R*8)  VALUE    = ???
C  INPUT : (I*4)  IMAX     = NUMBER OF VALUES IN ARR
C
C  OUTPUT: (I*4)  INDEX    = LOCATED POSITION
C
C          (L*4)  FOUND    = ???
C          (C*2)  OVER     = ???
C
C
C ROUTINES:
C          NONE
C
C
C
C AUTHOR:  HARVEY ANDERSON, UNIVERSITY OF STRATHCLYDE/JET
C          JA8.08
C          TEL. 0141-553-4196
C
C
C DATE:    16/05/97
C
C UNIX-IDL PORT: H.P.SUMMERS
C
C VERSION: 1.1 DATE: 10-07-97
C MODIFIED: HUGH SUMMERS, UNIVERSITY OF STRATHCLYDE
C     - PUT UNDER S.C.C.S. CONTROL
C
C
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      INTEGER             IMAX,        INDEX
      REAL*8              ARR(99),     VALUE
```

4.57 ccintp: Subroutine ccintp from library adas3xx

```
      SUBROUTINE CCINTP( ZDATA   , YDATA   , XDATA    , SRAY    ,
     &                   ITREF   , IEREF   , INREF    ,
     &  TVAL    , MAXEB   , MAXNE    , MAXTE   ,
     &                   EBRAY   , NERAY   , BMSTOP   , BMEMIS  ,
     &  BMDENS  , IECOUNT , INCOUNT  ,
     &                   F1      , NN      , AVALUE   , LEVEL   ,
     &  NLEVEL  , IEA     , INA      , ITA
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: CCDATA *********************
C
C  PURPOSE:  TO INTERPOLATE BETWEEN THE EFFECTIVE STOPPING
C   OR EMISSION COEFFICIENTS.
C
C  CALLING PROGRAM: ADAS312
C
C  SUBROUTINE:
C
C  INPUT : (R*8)  ZDATA(,)   =
C  INPUT : (R*8)  YDATA()    =
C  INPUT : (R*8)  XDATA()    =
C  INPUT : (R*8)  SRAY(,,)   =
C  INPUT : (I*4)  ITREF      =
C  INPUT : (I*4)  IEREF      =
C  INPUT : (I*4)  INREF      =
C  INPUT : (I*4)  TVAL       =
C  INPUT : (I*4)  MAXEB      =
C  INPUT : (I*4)  MAXNE      = MAXIMUM NUMBER OF DENSITIES
C  INPUT : (I*4)  MAXTE      =
C  INPUT : (R*8)  EBRAY()    =
C  INPUT : (R*8)  NERAY()    =
C  INPUT : (I*4)  BMSTOP     =
C  INPUT : (I*4)  BMEMIS     =
C  INPUT : (I*4)  BMDENS     =
C  INPUT : (I*4)  IECOUNT    =
C  INPUT : (I*4)  INCOUNT    =
C  INPUT : (R*8)  F1(,,,)    =
C  INPUT : (R*8)  NN(,,,)    =
C  INPUT : (R*8)  AVALUE     =
C  INPUT : (I*4)  LEVEL      =
C  INPUT : (I*4)  NLEVEL     =
C  INPUT : (I*4)  IEA()      =
C  INPUT : (I*4)  INA()      =
C  INPUT : (I*4)  ITA()      =
C
C          (I*4)  I          = GENERAL INTEGER VARIABLE
C          (I*4)  J          = GENERAL INTEGER VARIABLE
C
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4UNIT     ADAS      FETCH UNIT NUMBER FOR OUTPUT OF MESSAGES
C
C
C
C AUTHOR:  HARVEY ANDERSON, UNIVERSITY OF STRATHCLYDE/JET
C          JA8.08
```

```
C          TEL. 0141-553-4196
C
C
C DATE:    16/05/97
C
C UNIX-IDL PORT: H.P.SUMMERS
C
C VERSION: 1.1 DATE: 10-07-97
C MODIFIED: HUGH SUMMERS, UNIVERSITY OF STRATHCLYDE
C     - PUT UNDER S.C.C.S. CONTROL
C
C VERSION: 1.2                          DATE: 05-02-03
C MODIFIED: COSTANZA MAGGI, IPP GARCHING
C           - EXTENDED TO ALLOW WRITING BMP FILES
C
C-----------------------------------------------------------------------
      INTEGER             BMDENS,      BMEMIS,      BMSTOP
      INTEGER             IEA(MAXEB),  IECOUNT,     IEREF
      INTEGER             INA(MAXNE),  INCOUNT,     INREF
      INTEGER             ITA(MAXTE),  ITREF,       LEVEL,       MAXEB
      INTEGER             MAXNE,       MAXTE,       NLEVEL,      TVAL
      REAL*8              AVALUE,      EBRAY(MAXEB)
      REAL*8              F1(MAXEB,MAXNE,MAXTE,NLEVEL)
      REAL*8              NERAY(MAXNE)
      REAL*8              NN(MAXEB,MAXNE,MAXTE,NLEVEL)
      REAL*8              SRAY(MAXEB,MAXNE,MAXTE),  XDATA(MAXEB)
      REAL*8              YDATA(MAXNE),             ZDATA(MAXEB,MAXNE)
```

4.58 ccnse4: Subroutine ccnse4 from library adas3xx

```
       SUBROUTINE CCNSE4(A0,CA0,A,CA,RH,CRH,NMIN,NMAX,ARED,CARED,RHS,
     1CRHS,IR,ISH,JSH,KPF,NLIV,ILTXR)
C
C-----------------------------------------------------------------------
C
C ********** FORTRAN 77 ROUTINE : CCNSE4.F  **********
C
C PURPOSE : APPLIES MATRIX CONDENSATION TREATEMENT TO
C   THE ARRAYS WHICH ARE USED TO CONSTRUCT
C   THE COLLISIONAL RADIATIVE MATRIX.
C
C HISTORY : ROUTINE WAS ORIGINALLY WRITTEN BY H.P.SUMMERS.
C
C
C INPUT   :
C
C (R*8) A0 :
C (R*8) CAO :
C (R*8) A :
C (R*8) CA :
C (R*8) RH :
C (R*8) CRH :
C (I*4)   IR :  INDEX CORRESPONDING TO THE REPRESENTATIVE
C    LEVEL OF INTEREST.
C (I*4)   IRS :  SWITCH USED TO LOCATE THE DIAGONAL ELEMENTS
C    OF THE COLLISIONAL RADIATIVE MATRIX. THIS
C    ROUTINE DOES NOT CONTAIN THE C-R MATRIX BUT
C    ARRAYS WHICH ARE USED TO ASSEMBLE IT, FOR
C    A SINGLE SPIN SYSTEM WITH NO METASTABLES
C    IRS=0 .
C
C
C
C CONTACT : HARVEY ANDERSON
C   UNIVERSITY OF STRATHCLYDE
C   ANDERSON@PHYS.STRATH.AC.UK
C
C DATE : 26/02/98
C
C VERSION:      1.2                                     DATE:   21-10-99
C MODIFIED: RICHARD MARTIN
C               CHANGED HEXADECIMAL CONSTANTS TO Z'FFF00000' FORM.
C
C-----------------------------------------------------------------------
C
IMPLICIT REAL*8(A-H,O-Z)
C
      INTEGER             ILTXR(10,10,5),           IR,          ISH
      INTEGER             JSH,         KPF(1000),   NLIV(31),    NMAX
      INTEGER             NMIN
      REAL*8              A(1000),     A0,          ARED(160)
      REAL*8              CA(1000),    CA0,         CARED(160),  CRH
      REAL*8              CRHS,        RH,          RHS
```

4.59 ccnst7: Subroutine ccnst7 from library adas3xx

```
       SUBROUTINE CCNST7(NLREP,IR,I,V,E,EXE,EXS,K,ISG,CGBBA,ENL,ENL2,
     &                   EXPTE,EXPTS,KPF,ISG1,NMIN,NMAX,IMAX,
     &                   A0,CA0,A,CA,RH,CRH,
     &                   IOPT,NG,LG,NGL,LTG,FPG,NG1,LG1,
     &                   NGL1,LTG1,FPG1,ARL,ISARL,AC1,RHSC1,
     &                   C1,C2,C3,SCLA,NIMP,ZIMPA,FRIMPA,AMIMPA,
     &                   EXMTE )
C-----------------------------------------------------------------------
C
C       **********  FORTRAN 77 ROUTINE : CCNST7 **********
C
C       PURPOSE : ASSEMBLES ARRAYS USED TO CONSTRUCT THE
C                 COLLISIONAL-RADIATIVE MATRIX.
C
C       HISTORY : ROUTINE WAS ORIGINALLY WRITTEN BY H.P.SUMMERS.
C                 RESTRUCURED AND MODIFIED BY H.ANDERSON.
C
C       INPUT   :
C
C       (I*4) NLREP()    : ARRAY CONTAINING ALL OF THE REPRESENTATIVE
C                          LEVELS.
C       (I*4) IR         : COUNTER GIVEN AS INPUT WHICH RANGES FROM
C                          NMIN TO NMAX AND IS USED TO REFERENCE THE
C                          REPRESENTATIVE LEVEL.
C       (I*4) I          : REPRESENTATIVE LEVEL WHICH IS GIVEN BY
C                          NLREP(IR)
C       (R*8) V          : EFFECTIVE PRINCIPAL QUANTUM NUMBER FOR THE
C                          REPRESENTATIVE LEVEL I.
C       (R*8) E          : RECIPROCAL OF THE EFFECTIVE PRINCIPAL QUANTUM
C                          NUMBER SQUARED FOR THE REPRESENTATIVE LEVEL I.
C       (R*8) EXE        : VARIABLE ASSIGNED THE VALUE OF EXP(I/k*Te)
C                          ASSOCIATED WITH THE REPRESENTATIVE LEVEL I.
C       (R*8) EXS        : VARIABLE ASSIGNED THE VALUE OF EXP(I/k*TS)
C                          ASSOCIATED WITH THE REPRESENTATIVE LEVEL I.
C       (I*4) K          : VARIABLE ASSIGNED THE QUANTUM NUMBERS FOR
C                          THE REPRESENTATIVE LEVEL I. STORAGE OF THE
C                          NUMBERS ARE OF THE SAME FORMAT AS KPF().
C       (I*4) ISG        : MULTIPLICITY ?.
C
C       (R*8) CGBBA      : **** UNKNOWN ****
C       (R*8) ENL()      : ARRAY CONTAINING THE EFFECTIVE PRINCIPAL
C                          QUANTUM NUMBERS FOR EACH REPRESENTATIVE
C                          LEVEL.
C       (R*8) ENL2()     : ARRAY CONTAINING THE RECIPROCAL OF THE
C                          EFFECTIVE PRINCIPAL QUANTUM NUMBER
C                          SQUARED FOR EACH REPRESENTATIVE LEVEL.
C       (R*8) EXPTE()    : ARRAY CONTAINING THE VALUE OF EXP(I/k*Te)
C                          FOR EACH REPRESENTATIVE LEVEL FOR THE
C                          TRIPLETS.
C
C       (R*8) EXMTE()    : ARRAY CONTAINING THE VALUE OF EXP(I/k*Te)
C                          FOR EACH REPRESENTATIVE LEVEL FOR THE
C                          SINGLETS.
C       (R*8) EXPTS()    : ARRAY CONTAINING THE VALUE OF EXP(I/k*TS)
C                          FOR EACH REPRESENTATIVE LEVEL.
C       (I*4) KPF()      : ARRAY CONTAINING THE QUANTUM NUMBERS,
C                          N,l,L FROM NMIN TO NMAX, IN ORDER
C                          OF DECREASING BINDING ENERGY. EACH 32 BIT
C                          ELEMENT OF THE ARRAY IS USED TO STORE N,l
```

```
C                       AND L FOR EACH REPRESENTATIVE LEVEL. THE
C                       FIRST 10 BITS ARE USED TO STORE THE TOTAL
C                       ANGULAR MOMENTUM QUANTUM NUMBER. THE NEXT
C                       10 BITS ARE USED TO STORE THE ORBITAL
C                       QUANTUM NUMBER. THE LAST 12 BITS ARE USED
C                       TO STORE THE PRINCIPAL QUANTUM NUMBER.
C
C                       |<---------4 BYTE INTEGER--------->|
C                       |<---------32 BIT INTEGER--------->|
C                       |<----N----->|<----l--->|<---L---->|
C
C                     [ |.....N......|.....l....|.....L....|  ]
C
C                       BIT OPERATORS ARE THEN EMPLOYED TO
C                       INTEROGATE ARRAYS,E.G IAND,ISHFR,
C                       USING HEXIDECIMAL MASKS.
C
C       (I*4) ISG1      : *****UNKNOWN*****
C       (I*4) NMIN      : MINIMUM PRINCIPAL QUANTUM NUMBER OF THE
C                         RANGE WHICH CONTAINS THE REPRESENTATIVE
C                         LEVELS.
C
C       (I*4) NMAX      : MAXIMUM PRINCIPAL QUANTUM NUMBER OF THE
C                         RANGE WHICH CONTAINS THE REPRESENTATIVE
C                         LEVELS.
C       (I*4) IMAX      : THE MAXIMUM NUMBER OF REPRESENTATIVE
C                         LEVELS.
C
C       OUTPUT :
C
C       (R*8) A0        :
C       (R*8) CA0       :
C       (R*8) A         :
C       (R*8) CA        :
C       (R*8) RH        :
C       (R*8) CRH       :
C       (I*4) IOPT      : SWITCH USED TO DETERMINE IF CCNST7
C                         SHOULD ASSEMBLE ARRAYS USED TO
C                         CONSTRUCT THE COLLISIONAL-RADIATIVE
C                         MATRIX EXCLUDING SPIN CHANGING
C                         CROSS SECTIONS ( IOPT = 1 ) OR IF
C                         CCNST7 SHOULD ONLY CONTRUCT THE
C                         ARRAYS CONTAINING SPIN CHANGING
C                         CROSS SECTIONS ( IOPT GT 1 ).
C       (I*4) NG        :
C       (I*4) LG        :
C       (I*4) NGL       :
C       (I*4) LTG       :
C       (R*8) FPG       :
C       (I*4) NG1       :
C       (I*4) LG1       :
C       (I*4) NGL1      :
C       (I*4) iLTG1     :
C       (R*8) FPG1      :
C       (R*8) ARL       :
C       (I*4) ISARL     :
C       (R*8) AC1       :
C       (R*8) RHSC1     :
C       (R*8) C1()      : COEFFICIENTS OF THE QUANTUM DEFECT
C                         EXPANSION.
C       (R*8) C2()      : COEFFICIENTS OF THE QUANTUM DEFECT
```

```
C                           EXPANSION.
C       (R*8) C3()      : COEFFICIENTS OF THE QUANTUM DEFECT
C                           EXPANSION.
C       (R*8) SCLA      :
C       (I*4) NIMP      : NUMBER OF IMPURITIES IN THE PLASMA.
C       (R*8) ZIMPA     : ARRAY CONTAINING THE NUCLEAR CHARGE
C                           OF THE IMPURITIES IN THE PLASMA.
C       (R*8) FRIMPA    : IMPURITY FRACTIONS.
C       (R**) AMIMPA    : THE ATOMIC MASS OF EACH IMPURITY
C                                 WITHIN THE PLASMA.
C
C
C
C       ROUTINES:
C
C           ROUTINE        SOURCE         DESCRIPTION
C        ------------------------------------------------
C           GBBR            ADAS
C           GBB             ADAS
C           PYVR            ADAS
C           PYPR            ADAS
C           PYIPHE          ADAS
C           RQVNEW          ADAS
C           RQLNEW          ADAS
C           RQBNEW          ADAS
C           GHNLV           ADAS
C           GHNLE           ADAS
C           COLINT          ADAS
C           RQINEW          ADAS
C           PHOTO2          ADAS
C           NDIEL           ADAS
C           RNDEGV          ADAS
C           WIG6J           ADAS
C           OVLP            ADAS
C           COLEXC          ADAS
C           .....
C           .....
C           .....
C
C
C       CONTACT : HARVEY ANDERSON
C                 UNIVERSITY OF STRATHCLYDE
C                 ANDERSON@BARWANI.PHYS.STRATH.AC.UK
C
C       DATE    : 2/2/98
C
C
C
C       WARNING!!! : CODE IS UNDER DEVELOPMENT
C
C                    HARVEY ANDERSON
C                    UNIVERSITY OF STRATHCLYDE
C
C VERSION:      1.2                                     DATE:   21-10-99
C MODIFIED: RICHARD MARTIN
C               CHANGED HEXADECIMAL CONSTANTS TO Z'FFF00000' FORM.
C               CORRECTED ARRAY INDEXING PROBLEM
C
C VERSION : 1.3
C MODIFIED: Martin O'Mullane
C DATE    : 3-6-2000
```

```
C           iLTG1 is a scalar here but is passed from start7
C           as an integer array. It should be LTG1 not iLTG1.
C           COR should be 20 not 6 - seel NDIEL subroutine.
C           CORS should be an array not a scalar.
C
C VERSION : 1.4
C DATE    : 18-11-2004
C MODIFIED: Martin O'Mullane
C            - Align with Harvey Anderson's last version.
C            - Make implicit none and remove unecessary code.
C
C VERSION : 1.5
C DATE    : 24-02-2005
C MODIFIED: Martin O'Mullane
C            - Make implicit none and remove unecessary code.
C
C-----------------------------------------------------------------------
      INTEGER             I,           IMAX,        IOPT,        IR
      INTEGER             ISARL(80),   ISG,         ISG1,        K
      INTEGER             KPF(1000),   LG,          LG1,         LTG(5)
      INTEGER             LTG1(5),     NG,          NG1,         NGL
      INTEGER             NGL1,        NIMP,        NLREP(80),   NMAX
      INTEGER             NMIN
      REAL*8              A(1000),     A0,          AC1
      REAL*8              AMIMPA(10),  ARL(80),     C1(5,5,3)
      REAL*8              C2(5,5,3),   C3(5,5,3),   CA(1000),    CA0
      REAL*8              CGBBA(800,3),             CRH,         E
      REAL*8              ENL(1000),   ENL2(1000),  EXE
      REAL*8              EXMTE(1000), EXPTE(1000), EXPTS(1000), EXS
      REAL*8              FPG(5),      FPG1(5),     FRIMPA(10),  RH
      REAL*8              RHSC1,       SCLA(240,8), V
      REAL*8              ZIMPA(10)
```

4.60 ccout0: Subroutine ccout0 from library adas3xx

```
      SUBROUTINE CCOUT0( OFILE  , OUNIT  , IZ     , IECOUNT , INCOUNT ,
     &  TEREF  , EBRAY  , NERAY  , XRAY    ,
     &                    IEA    , INA    , ITREF  ,
     &  ITCOUNT,
     &                    EBREF  , NEREF  , TERAY  ,
     &                    MAXTE  , MAXNE  , MAXEB  ,
     &  ITA    , IEREF  , INREF  ,
     &                    DATE   , TITLE  , BMSTOP , BMEMIS  , BMDENS ,
     &  F1     , NN     , AVALUE , NLEVEL  , LEVEL  ,
     &                    INFILE )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: CCOUT0 *********************
C
C  PURPOSE:  TO WRITE OUTPUT TO DATA FORMAT ADF21/22 SPECIFICATIONS.
C
C  CALLING PROGRAM: ADAS312
C
C  SUBROUTINE:
C
C  INPUT : (R*8)   TEREF     = THE REFERENCE TARGET TEMPERATURE (eV).
C  INPUT : (R*8)   EBREF     = THE REFERENCE NEUTRAL BEAM ENERGY
C   (eV/amu).
C  INPUT : (R*8)   NEREF     = THE REFERENCE TARGET ELECTRON DENSITY.
C   (CM-**3)
C  INPUT : (R*8)   TERAY()   = ARRAY CONTAINING THE TARGET TEMPERATURE.
C  INPUT : (R*8)   EBRAY()   = ARRAY CONTAINING THE NEUTRAL BEAM
C   ENERGY ( eV/amu ).
C  INPUT : (R*8)   NERAY()   =  ARRAY CONTAINING THE TARGET DENSITY.
C  INPUT : (R*8)   XRAY()    = ARRAY CONTAINING EITHER THE BEAM STOPPING
C   OR EMISSION COEFFICIENTS (CM**3/S).
C
C  INPUT : (R*8)   IZ        = THE EFFECTIVE NUCLEAR CHARGE OF THE
C   TARGET PLASMA.
C  INPUT : (I*4)   ITCOUNT   = THE NUMBER OF TARGET TEMPERATURES.
C  INPUT : (I*4)   IECOUNT   = THE NUMBER OF BEAM ENERGIES.
C  INPUT : (I*4)   INCOUNT   = THE NUMBER OF TARGET DENSITIES.
C  INPUT : (I*4)   ITA()     = ARRAY CONTAINING THE INDEX VALUES
C   OF THE TEMPERATURE ARRAY TERAY().
C  INPUT : (I*4)   ITREF     = THE INDEX VALUE AT WHICH THE REFERENCE
C   TEMPERATURE CORRESPONDS TO IN THE ARRAY
C   TERAY.
C  INPUT : (I*4)   IEA()     = ARRAY CONTAINING THE INDEX VALUES
C   OF THE ENERGY ARRAY EBRAY().
C  INPUT : (I*4)   IEREF     = THE INDEX VALUE AT WHICH THE REFERENCE
C   ENERGY CORRESPONDS TO IN THE ARRAY
C   EBRAY().
C  INPUT : (I*4)   INA()     = ARRAY CONTAINING THE INDEX VALUES
C   OF THE DENSITY ARRAY NERAY().
C  INPUT : (I*4)   INREF     = THE INDEX VALUE AT WHICH THE REFERENCE
C   DENSITY CORRESPONDS TO IN THE ARRAY
C   NERAY.
C  INPUT : (I*4)   MAXTE     = THE MAXIMUM NUMBER OF TARGET TEMPERATURES.
C  INPUT : (I*4)   MAXNE     = THE MAXIMUM NUMBER OF TARGET DENSITIES.
C  INPUT : (I*4)   MAXEB     = THE MAXIMUM NUMBER OF BEAM ENERGIES.
C  INPUT : (I*4)   NUM       = THE NUMBER OF ELEMENT SYMBOLS CONTAINED
C   IN THE ARRAY SPEC().
C
C
```

```
C (CHR) OFILE        = OUTPUT FILENAME.
C (CHR) SPEC()       = ARRAY CONTAING THE CHEMICAL SYMBOLS
C   OF THE ELEMENTS UP TO THE FIRST
C   PERIOD.
C (CHR) HEADER       = STRING CONTAINING THE ADAS VERSION,
C   THE EXECUTING PROGRAM AND DATE.
C
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4UNIT     ADAS      FETCH UNIT NUMBER FOR OUTPUT OF MESSAGES
C          CCFIND     ADAS      ???
C
C
C
C AUTHOR:  HARVEY ANDERSON, UNIVERSITY OF STRATHCLYDE/JET
C          JA8.08
C          TEL. 0141-553-4196
C
C
C DATE:    16/05/97
C
C UNIX-IDL PORT: H.P.SUMMERS
C
C VERSION: 1.1 DATE: 10-07-97
C MODIFIED: HUGH SUMMERS, UNIVERSITY OF STRATHCLYDE
C     - PUT UNDER S.C.C.S. CONTROL
C
C VERSION: 1.2 DATE: 28-07-97
C MODIFIED: HARVEY ANDERSON, UNIVERSITY OF STRATHCLYDE
C     - INCLUDED PROGRAM NAME IN THE OUTPUT FILE.
C     - ADDED ADDITIONAL INFORMATION AT THE BOTTOM OF
C       THE ADF21/22 TYPE FILE.
C
C VERSION: 1.3 RICHARD MARTIN DATE: 1-11-97
C               - CORRECTED SYNTAX ERROR IN GETENV COMMAND.
C
C VERSION: 1.4 HARVEY ANDERSON DATE: 17-03-99
C - ADDED EXTRA OUTPUT OPTION.
C
C VERSION: 1.5 RICHARD MARTIN DATE: 28-04-99
C         - ADDED MISSING COMMA IN FORMAT STATEMENT 2013.
C
C VERSION: 1.6 HARVEY ANDERSON DATE: 08/09/99
C         - EXTENDED THE ARRAY CONTAINING THE CHEMICAL ELEMENT
C      SYMBOL TO INCLUDE ALL SPECIES UP TO ZN.
C
C VERSION: 1.7 COSTANZA MAGGI           DATE: 05-02-03
C               - EXTENDED TO ALLOW WRITING BMP FILES
C
C VERSION: 1.8 Martin O'Mullane  DATE: 06-02-2003
C   Improve information at bottom of output file.
C
C-----------------------------------------------------------------------
      CHARACTER*24        DATE
      CHARACTER*80        INFILE,      OFILE
      CHARACTER*24        TITLE
      INTEGER             BMDENS,      BMEMIS,      BMSTOP
      INTEGER             IEA(MAXEB),  IECOUNT,     IEREF
      INTEGER             INA(MAXNE),  INCOUNT,     INREF
```

```
INTEGER             ITA(MAXTE),  ITCOUNT,     ITREF,       IZ
INTEGER             LEVEL,       MAXEB,       MAXNE,       MAXTE
INTEGER             NLEVEL,      OUNIT
REAL*8              AVALUE,      EBRAY(MAXNE),             EBREF
REAL*8              F1(MAXEB,MAXNE,MAXTE,NLEVEL)
REAL*8              NERAY(MAXNE),             NEREF
REAL*8              NN(MAXEB,MAXNE,MAXTE,NLEVEL)
REAL*8              TERAY(MAXTE),             TEREF
REAL*8              XRAY(MAXEB,MAXNE,MAXTE)
```

4.61 ccsort: Subroutine ccsort from library adas3xx

```
      SUBROUTINE CCSORT( XA   , IA   , N )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: CCSORT *********************
C
C  PURPOSE:  TO SORT AN ARRAY SO THAT XA IS INCREASING ORDER.
C
C  CALLING PROGRAM: ADAS312
C
C  SUBROUTINE:
C
C  I/O   : (R*8) XA()     = X-VALUES INITIAL THEN SORTED
C  I/O   : (I*4) IA()     = I-VALUES INITIAL THEN SORTED
C  INPUT : (I*4) N        = NUMBER OF VALUES
C
C          (I*4) N1       = GENERAL INTEGER VARIABLE
C          (I*4) I        = GENERAL INTEGER VARIABLE
C          (I*4) I1       = GENERAL INTEGER VARIABLE
C          (I*4) J        = GENERAL INTEGER VARIABLE
C          (I*4) ISWAP    = GENERAL INTEGER VARIABLE
C          (R*8) SWAP     = GENERAL REAL VARIABLE
C
C
C ROUTINES:
C          NONE
C
C
C
C AUTHOR:  HARVEY ANDERSON, UNIVERSITY OF STRATHCLYDE/JET
C          JA8.08
C          TEL. 0141-553-4196
C
C
C DATE:    16/05/97
C
C UNIX-IDL PORT: H.P.SUMMERS
C
C VERSION: 1.1 DATE: 10-07-97
C MODIFIED: HUGH SUMMERS, UNIVERSITY OF STRATHCLYDE
C     - PUT UNDER S.C.C.S. CONTROL
C
C
C
C-----------------------------------------------------------------------
      INTEGER             IA(*),       N
      REAL*8              XA(*)
```

4.62 cdaval: Subroutine cdaval from library adas3xx

```
        SUBROUTINE CDAVAL(IUNIT,DSNAME,UN,UL,LN,LL,MULTI,AVALUE)
C-----------------------------------------------------------------------
C
C   *********   FORTRAN77 SUBROUTINE: CDAVAL *********
C
C   PURPOSE:  EXTRACTS A-VALUE FROM AN ADF04 TYPE FILE GIVEN
C   THE QUANTUM NUMBERS DESCRIBING THE INITIAL AND
C   FINAL CONFIGURATION.
C
C   CALLING PROGRAM:  ADAS313
C
C INPUT  :
C
C (I*4) UN : UPPER PRINCIPAL QUANTUM
C   NUMBER.
C (I*4) UL : UPPER TOTAL L QUANTUM
C   NUMBER.
C (I*4) LN : LOWER PRINCIPAL QUANTUM
C   NUMBER.
C (I*4) LL : LOWER TOTAL L QUANTUM
C   NUMBER.
C (I*4) MULTI : MULTIPLICITY.
C (I*4) IUNIT : STREAM NUMBER FOR ADF04
C   TYPE FILE.
C (CHR) DSNAME : ADF04 FILENAME INCLUDING
C   THE PATH.
C
C OUPUT :
C (R*8) AVALUE : TRANSITION PROBABILITY.
C
C
C GENERAL :
C
C (I*4) IA() : INDEX ARRAY.
C (I*4) NB() : PRINCIPAL QUANTUM NUMBER.
C (I*4) NA() : PRINCIPAL QUANTUM NUMBER.
C (I*4) MULTA() : MULTIPLICITY.
C (I*4) LTA() : TOTAL L QUANTUM NUMBER.
C (I*4) I : GENERAL COUNTER.
C (I*4) J : GENERAL COUNTER.
C (I*4) NDLEVELS : NUMBER OF LEVELS CONTAINED
C   IN THE ADF04 FILE.
C (I*4) UREF : INDEX REFERENCING THE UPPER
C   LEVEL.
C (I*4) LREF : INDEX REFERENCING THE LOWER
C   LEVEL.
C (I*4) UINDX : INDEX OF THE UPPER LEVEL
C   CONTAINED IN THE ADF04 TYPE
C   FILE.
C (I*4) LINDX : INDEX OF THE LOWER LEVEL
C   CONTAINED IN THE ADF04 TYPE
C   FILE.
C
C
C  CONTACT : HARVEY ANDERSON
C        UNIVERSITY OF STRATHCLYDE
C           ANDERSON@PHYS.STRATH.AC.UK
C
C  DATE    : 04/29/98
```

```
C
C VERSION: 1.1 DATE: 16-03-99
C MODIFIED: RICHARD MARTIN
C - PUT UNDER SCCS CONTROL.
C
C VERSION: 1.2 DATE: 15-10-99
C MODIFIED: RICHARD MARTIN
C REMOVED SUPERFLUOUS WRITE STATEMAENT.
C
C-----------------------------------------------------------------------
C
      CHARACTER*80        DSNAME
      INTEGER             IUNIT,       LL,          LN,          MULTI
      INTEGER             UL,          UN
      REAL*8              AVALUE
```

4.63 cddata: Subroutine cddata from library adas3xx

```
       SUBROUTINE CDDATA( TERAY   ,   NERAY ,  EBRAY      ,IZ      ,
     &   INFILE  ,   INUNIT    ,  MAXNE      ,MAXTE   ,
     &                    MAXEB   ,   INCOUNT   ,  ITCOUNT    ,IECOUNT ,
     &    EBREF   ,   TEREF     ,  NEREF      ,INA     ,
     &   IEA     ,   ITA       ,  GCRC1      ,GCRC2   ,
     &   GCRC3   ,   GCRC4     ,  GCRC5      ,GCRC6   ,
     &   GCRC7   ,   GCRC8     ,  GCRC9      ,NPQN    ,
     &   LTPQN   ,   NSPIN     ,  BNL        ,NNBNL   ,
     &   FI      ,   FII       ,  FIII       , CHOICE )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: CDDATA *********************
C
C       PURPOSE:  TO  FETCH  DATA  FROM  BUNDLE-NL POPULATION
C  STRUCTURE FILES OF TYPE ADF26.
C
C       CALLING ROUTINE : ADAS312
C
C INPUT :
C
C (CHR) INFILE : FILENAME FOR WHICH DATA HAS
C   TO BE EXTRACTED FROM.
C (I*4) INUNIT : FILENAME STREAM.
C (I*4) MAXNE : MAXIUM NUMBER OF DENSITIES
C (I*4) MAXEB : MAXIMUM NUMBER OF BEAM
C   ENERGIES.
C (I*4) MAXTE : MAXIMUM NUMBER OF TARGET
C   TEMPERATURES
C (I*4) NPQN : PRINCIPAL QUANTUM NUMBER.
C (I*4) LTPQN : TOTAL ANGULAR MOMENTUM
C   QUANTUM NUMBER.
C (I*4)   CHOICE : SWITCH TO SELECT COUPLING
C   OR EMISSION COEFFICIENTS.
C   CHOICE = 1 EXTRACT COUPLING
C   COEFFICIENTS.
C   CHOICE = 2 EXTRACT EMISSION
C   COEFFICIENTS.
C
C OUTPUT  :
C
C (R*8)   TERAY() : TARGET TEMPERATURES (eV).
C (R*8) NERAY() : ELECTRON DENSITY ( cm-3).
C (R*8) EBRAY() : NEUTRAL BEAM ENERGY (eV/amu).
C (R*8) TEREF : REFERENCE TEMPERATURE ( eV ).
C (R*8) NEREF : REFERENCE DENSITY ( cm-3).
C (R*8) GCRC1() : CROSS COUPLING COEFF.(cm-3s-1)
C (R*8) GCRC2() : CROSS COUPLING COEFF.(cm-3s-1)
C (R*8) GCRC3() : CROSS COUPLING COEFF.(cm-3s-1)
C (R*8) GCRC4() : CROSS COUPLING COEFF.(cm-3s-1)
C (R*8) GCRC5() : CROSS COUPLING COEFF.(cm-3s-1)
C (R*8) GCRC6() : CROSS COUPLING COEFF.(cm-3s-1)
C (R*8) GCRC7() : CROSS COUPLING COEFF.(cm-3s-1)
C (R*8) GCRC8() : CROSS COUPLING COEFF.(cm-3s-1)
C (R*8) GCRC9() : CROSS COUPLING COEFF.(cm-3s-1)
C (R*8) FI() : CONTRIBUTION TO POPULATING THE
C   SINGLETS RELATIVE TO THE
C   GROUND STATE DUE TO EXCITATION
C (R*8) FII() : CONTRIBUTION TO POPULATING THE
C   SINGLETS RELATIVE TO THE 2(1)S
```

```
C   METASTABLE.
C (R*8) FIII() : CONTRIBUTION TO POPULATING THE
C   TRIPLETS RELATIVE TO THE 2(3)S
C   METASTABLE.
C (R*8) BNL() : SAHA-BOLTZMANN B-FACTOR
C (R*8)   NNBNL() : THE PRODUCT OF THE RELATIVE
C   POPULATION OF A PARTICULAR
C   LEVEL TO THE FIRST IONISATION
C   STAGE AND THE RECIPROCAL OF
C   THE SAHA-BOLTZMANN B-FACTOR.
C
C (I*4) INA() : REFERENCE ARRAY FOR DENSITY.
C (I*4) IEA() : REFERENCE ARRAY FOR ENERGY.
C (I*4) ITA() : REFERENCE ARRAY FOR TEMPERATURE.
C (I*4)   NSPIN : NUMBER OF SPIN SYSTEMS.
C (I*4) INCOUNT : NUMBER OF TARGET DENSITIES.
C (I*4) IECOUNT : NUMBER OF BEAM ENERGIES.
C (I*4) ITCOUNT : NUMBER OF TEMPERATURES.
C (I*4) IZ : NUCLEAR CHARGE.
C
C
C  ADDITIONAL ROUTINES:
C
C          ROUTINE       SOURCE            BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          CCFIND       ADAS312      ISOLATE DATA IN ADF26 TYPE FILE.
C
C
C
C   CONTACT :   HARVEY ANDERSON
C            UNIVERSITY OF STRATHCLYDE
C            ANDERSON@PHYS.STRATH.AC.UK
C
C   DATE    :   23/4/98  ( FIRST VERSION )
C
C
C VERSION: 1.1 DATE: 16-03-99
C MODIFIED: RICHARD MARTIN
C - PUT UNDER SCCS CONTROL.
C
C VERSION: 1.2 DATE: 13-10-99
C MODIFIED: Martin O'Mullane
C - With certain compilers array dimension variables
C                must be declared before they are used. Move the
C                integer declerations before the array definitions.
C
C-----------------------------------------------------------------------
      CHARACTER*80        INFILE
      INTEGER             CHOICE,      IEA(MAXNE),  IECOUNT
      INTEGER             INA(MAXNE),  INCOUNT,     INUNIT
      INTEGER             ITA(MAXTE),  ITCOUNT,     IZ,          LTPQN
      INTEGER             MAXEB,       MAXNE,       MAXTE,       NPQN
      INTEGER             NSPIN
      REAL*8              BNL(MAXEB,MAXNE,MAXTE,NSPIN+1)
      REAL*8              EBRAY(MAXEB),             EBREF
      REAL*8              FI(MAXEB,MAXNE,MAXTE,NSPIN+1)
      REAL*8              FII(MAXEB,MAXNE,MAXTE,NSPIN+1)
      REAL*8              FIII(MAXEB,MAXNE,MAXTE,NSPIN+1)
      REAL*8              GCRC1(MAXEB,MAXNE,MAXTE)
      REAL*8              GCRC2(MAXEB,MAXNE,MAXTE)
      REAL*8              GCRC3(MAXEB,MAXNE,MAXTE)
```

```
REAL*8              GCRC4(MAXEB,MAXNE,MAXTE)
REAL*8              GCRC5(MAXEB,MAXNE,MAXTE)
REAL*8              GCRC6(MAXEB,MAXNE,MAXTE)
REAL*8              GCRC7(MAXEB,MAXNE,MAXTE)
REAL*8              GCRC8(MAXEB,MAXNE,MAXTE)
REAL*8              GCRC9(MAXEB,MAXNE,MAXTE), NERAY(MAXNE)
REAL*8              NEREF,       NNBNL(MAXEB,MAXNE,MAXTE,NSPIN+1)
REAL*8              TERAY(MAXTE),             TEREF
```

4.64 cdintp: Subroutine cdintp from library adas3xx

```
SUBROUTINE CDINTP(CHOICE    ,OPTION   ,IECOUNT  ,INCOUNT  ,
     &   MAXNE     ,MAXTE    ,MAXEB    ,ZDATA    ,
     &   GCRC1     ,GCRC2    ,GCRC3    ,GCRC4    ,
     &   GCRC5     ,GCRC6    ,GCRC7    ,GCRC8    ,
     &   GCRC9     ,IEREF    ,INREF    ,ITREF    ,
     &   TVAL      ,RELMET   ,FI ,INA      ,
     &   IEA       ,FII      ,FIII     ,NSPIN    ,
     &   NNBNL     ,AVALUE   ,NERAY    ,ITA      ,
     &   EBRAY     ,XDATA    ,YDATA    ,MULTI  )
C
C-----------------------------------------------------------------------
C
C   ****** FORTRAN77 ROUTINE: CDINTP  ********
C
C   PURPOSE: INTERPOLATE BETWEEN THE EFFECTIVE CLOSE COUPLING
C  COEFFICIENTS OF EFFECTIVE EMISSION COEFFICIENTS.
C
C
C CALLING PROGRAM : ADAS313
C
C INPUT  :
C
C (R*8) FI() : THE CONTRIBUTION FROM
C   EXCITATION TO POPULATE A
C   LEVEL RELATIVE TO THE
C   GROUND STATE METASTABLE
C (R*8) FII() : THE CONTRIBUTION FROM
C   EXCITATION TO POPULATE A
C   LEVEL RELATIVE TO THE
C   2(1)S METASTABLE.
C (R*8) FIII() : THE CONTRIBUTION FROM
C   EXCITATION TO POPULATE A
C   LEVEL RELATIVE TO THE
C   2(3)S METASTABLE.
C (R*8)   GCRC1 : C-R COUPLING COEFFICIENT.
C (R*8)   GCRC2 : C-R COUPLING COEFFICIENT.
C (R*8)   GCRC3 : C-R COUPLING COEFFICIENT.
C (R*8)   GCRC4 : C-R COUPLING COEFFICIENT.
C (R*8)   GCRC5 : C-R COUPLING COEFFICIENT.
C (R*8)   GCRC6 : C-R COUPLING COEFFICIENT.
C (R*8)   GCRC7 : C-R COUPLING COEFFICIENT.
C (R*8)   GCRC8 : C-R COUPLING COEFFICIENT.
C (R*8)   GCRC9 : C-R COUPLING COEFFICIENT.
C (R*8) NNBNL() : THE PRODUCT OF THE RELATIVE
C                   POPULATION OF A PARTICULAR
C              LEVEL TO THE FIRST IONISATION
C      STAGE AND THE RECIPROCAL OF
C              THE SAHA-BOLTZMANN B-FACTOR.
C (R*8)   AVALUE : TRANSITION PROBABILTY.
C (R*8) NERAY() : ARRAY CONTAINING THE
C   ELECTRON DENSITY (CM-3).
C (R*8) EBRAY() : ARRAY CONTAINING THE BEAM
C           ENERGIES (eV/amu)
C (I*4) CHOICE : SWITCH USED TO INDICATE IF
C   COUPLING COEFFICIENCTS OR
C   EMISSION DATA IS BEING
C   HANDLED. CHOICE EQ 1 THEN
C   COUPLING DATA. CHOICE EQ 2
C        THEN EMISSION DATA.
```

```
C (I*4) OPTION : SIWTCH USED TO SELECT THE
C   ACTUAL CROSS COUPLING
C   COEFFICIENT.
C (I*4) IECOUNT : NUMBER OF BEAM ENERGIES.
C (I*4) INCOUNT : NUMBER OF TARGET DENSITIES.
C (I*4) MAXNE : THE MAXIMUM NUMBER OF
C   TARGET DENSITIES
C (I*4) MAXTE : MAXIMUM NUMBER OF
C   TEMPERATURES.
C (I*4) MAXEB : THE MAXIMUM NUMBER OF
C   BEAM ENERGIES
C (I*4) IEREF : INDEX TO THE REFERENCE
C   BEAM ENERGY.
C (I*4) INREF : INDEX TO THE REFERENCE
C   TARGET DENSITY
C (I*4) ITREF : INDEX TO THE REFERENCE
C   TEMPERATURE.
C (I*4) TVAL : ARRAY INDEX OF TEMPERATURE
C   AT WHICH DATA HAS TO BE
C   GENERATED.
C (I*4)   RELMET : SPECIFIES THE RELATIVE METASTABLE.
C         RELMET.EQ.1 1S2(1)S METASTABLE
C         RELMET.EQ.2 1S2S(1)S METASTABLE
C         RELMET.EQ.3 1S2S(3)S METASTABLE
C (I*4) INA() : ARRAY CONTAINING THE INDEXES
C   FOR THE DENSITY ARRAY.
C (I*4) IEA() : ARRAY CONTAINING THE INDEXES
C   FOR THE BEAM ENERGY ARRAY.
C (I*4) ITA() : ARRAY CONTAINING THE INDEXES
C   FOR THE TEMPERATURE ARRAY.
C (I*4)   NSPIN : NUMBER OF SPIN SYSTEMS.
C
C OUTPUT :
C
C (R*8) ZDATA() : ARRAY CONTAINING THE DERIVED
C   DATA AS A FUNCTION OF BEAM
C   ENERGY AND TARGET DENSITY
C   FOR A SELECTED TEMPERATURE.
C (R*8) YDATA() : ARRAY CONTAINING THE
C   ELECTRON DENSITY (CM-3).
C (R*8) XDATA() : ARRAY CONTAINING THE BEAM
C           ENERGIES (eV/amu)
C
C CONTACT : HARVEY ANDERSON
C   UNIVERSITY OF STRATHCLYDE
C   ANDERSON@PHYS.STRATH.AC.UK
C
C DATE    : 07/5/98
C  (FIRST VERSION)
C
C
C VERSION: 1.1 DATE: 16-03-99
C MODIFIED: RICHARD MARTIN
C - PUT UNDER SCCS CONTROL.
C
C-----------------------------------------------------------------------
C

      INTEGER             CHOICE,      IEA(MAXEB),  IECOUNT,     IEREF
      INTEGER             INA(MAXNE),  INCOUNT,     INREF
      INTEGER             ITA(MAXTE),  ITREF,       MAXEB,       MAXNE
```

```
INTEGER             MAXTE,       MULTI,       NSPIN,       OPTION
INTEGER             RELMET,      TVAL
REAL*8              AVALUE,      EBRAY(MAXEB)
REAL*8              FI(MAXEB,MAXNE,MAXTE,NSPIN+1)
REAL*8              FII(MAXEB,MAXNE,MAXTE,NSPIN+1)
REAL*8              FIII(MAXEB,MAXNE,MAXTE,NSPIN+1)
REAL*8              GCRC1(MAXEB,MAXNE,MAXTE)
REAL*8              GCRC2(MAXEB,MAXNE,MAXTE)
REAL*8              GCRC3(MAXEB,MAXNE,MAXTE)
REAL*8              GCRC4(MAXEB,MAXNE,MAXTE)
REAL*8              GCRC5(MAXEB,MAXNE,MAXTE)
REAL*8              GCRC6(MAXEB,MAXNE,MAXTE)
REAL*8              GCRC7(MAXEB,MAXNE,MAXTE)
REAL*8              GCRC8(MAXEB,MAXNE,MAXTE)
REAL*8              GCRC9(MAXEB,MAXNE,MAXTE), NERAY(MAXNE)
REAL*8              NNBNL(MAXEB,MAXNE,MAXTE,NSPIN+1)
REAL*8              XDATA(MAXEB),             YDATA(MAXNE)
REAL*8              ZDATA(MAXEB,MAXNE)
```

4.65 cdsum: Subroutine cdsum from library adas3xx

```
       SUBROUTINE CDSUM(  INUNIT   ,INFILE    ,TERAY   ,ITCOUNT,
     &   INCOUNT  ,IECOUNT   ,TEREF   ,NEREF  ,
     &   EBREF    ,ITREF     ,IEREF   ,INREF  ,
     &   ITA       ,IEA      ,INA     ,EBRAY  ,
     &   NERAY     , MAXTE   ,MAXEB   ,MAXNE  )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: CDSUM *********************
C
C       PURPOSE:  TO  FETCH  A SUMMARY OF THE DATA CONTAINED IN THE
C   BUNDLE-NL POPULATION STRUCTURE FILES OF TYPE ADF26.
C
C       CALLING ROUTINE : ADAS313
C
C INPUT :
C
C (CHR) INFILE : FILENAME FOR WHICH DATA HAS
C   TO BE EXTRACTED FROM.
C (I*4) INUNIT : FILENAME STREAM.
C (I*4) MAXNE : MAXIUM NUMBER OF DENSITIES
C (I*4) MAXEB : MAXIMUM NUMBER OF BEAM
C   ENERGIES.
C (I*4) MAXTE : MAXIMUM NUMBER OF TARGET
C   TEMPERATURES
C
C OUTPUT  :
C
C (R*8)   TERAY() : TARGET TEMPERATURES (eV).
C (R*8) NERAY() : ELECTRON DENSITY ( cm-3).
C (R*8) EBRAY() : NEUTRAL BEAM ENERGY (eV/amu).
C (R*8) TEREF : REFERENCE TEMPERATURE ( eV ).
C (R*8) NEREF : REFERENCE DENSITY ( cm-3).
C (R*8)   EBREF : REFERNCE ENERGY ( eV amu-1 ).
C (I*4) INA() : REFERENCE ARRAY FOR DENSITY.
C (I*4) IEA() : REFERENCE ARRAY FOR ENERGY.
C (I*4) ITA() : REFERENCE ARRAY FOR TEMPERATURE.
C (I*4) INREF : ARRAY INDEX OF REFERENCE DENSITY.
C (I*4) ITREF : ARRAY INDEX OF REFERENCE TEMP.
C (I*4) IEREF : ARRAY INDEX OF REFERENCE ENERGY.
C (I*4) INCOUNT : NUMBER OF TARGET DENSITIES.
C (I*4) IECOUNT : NUMBER OF BEAM ENERGIES.
C (I*4) ITCOUNT : NUMBER OF TEMPERATURES.
C
C
C  ADDITIONAL ROUTINES:
C
C          ROUTINE       SOURCE            BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          CCFIND       ADAS312      ISOLATE DATA IN ADF26 TYPE FILE.
C          CCFILL       ADAS312      USED TO ORDER AND SORT ARRAYS.
C          CCSORT       ADAS312      USED TO ORDER AND SORT ARRAYS.
C
C
C
C   CONTACT :   HARVEY ANDERSON
C            UNIVERSITY OF STRATHCLYDE
C            ANDERSON@PHYS.STRATH.AC.UK
C
C   DATE    :   07/05/98  ( FIRST VERSION )
```

```
C
C VERSION: 1.1 DATE: 16-03-99
C MODIFIED: RICHARD MARTIN
C - PUT UNDER SCCS CONTROL.
C
C VERSION: 1.2                                    DATE: 13-10-99
C MODIFIED: Martin O'Mullane
C   - With certain compilers array dimension variables
C must be declared before they are used. Move the
C integer declerations before the array definitions.
C
C-----------------------------------------------------------------------
C
      CHARACTER*80        INFILE
      INTEGER             IEA(MAXNE),  IECOUNT,     IEREF
      INTEGER             INA(MAXNE),  INCOUNT,     INREF,       INUNIT
      INTEGER             ITA(MAXTE),  ITCOUNT,     ITREF,       MAXEB
      INTEGER             MAXNE,       MAXTE
      REAL*8              EBRAY(MAXEB),             EBREF
      REAL*8              NERAY(MAXNE),             NEREF
      REAL*8              TERAY(MAXTE),             TEREF
```

4.66 ceecon: Subroutine ceecon from library adas3xx

```
      SUBROUTINE CEECON( INTYP , OUTTYP, IEVAL, EIN,
     &                   AMDON , AMREC , EOUT
     &                 )
C-----------------------------------------------------------------------
C  ****************** FORTRAN77 SUBROUTINE: CEECON *********************
C
C  PURPOSE: TO CONVERT AN ARRAY OF ENERGIES INTO A  SPECIFIED FORM.
C
C  CALLING PROGRAM: ADAS314
C  SUBROUTINE:
C  INPUT :    (I*4)  INTYP   = 1 => 'EIN(array)' UNITS: eV (Donor temp.)
C                            = 2 => 'EIN(array)' UNITS: eV (Recvr temp.)
C                            = 3 => 'EIN(array)' UNITS: eV/AMU  (ENERGY)
C  INPUT :    (I*4)  OUTTYP  = 1 => 'EOUT(array)' UNITS: eV (Donor temp)
C                            = 2 => 'EOUT(array)' UNITS: eV (Recvr temp)
C                            = 3 => 'EOUT(array)' UNITS: EV/AMU (ENERGY)
C  INPUT :    (I*4)  IEVAL   = NO. OF ENERGIES IN EIN(array)
C  INPUT :    (R*8)  EIN()   = INPUT  ENERGIES (STATED UNITS)
C  INPUT :    (R*8)  AMDON   = DONOR MASS NUMBER
C  INPUT :    (R*8)  AMREC   = RECEIVER MASS NUMBER
C  OUTPUT:    (R*8)  EOUT()  = OUTPUT ENERGIES (STATED UNITS)
C             (I*4)  I       = GENERAL USE
C             (R*8)  ECONV() = ENERGY CONVERSION PARAMETERS
C ROUTINES:  NONE
C NOTE:
C            ENERGY CONVERSION PARAMETERS:
C            INTYP = 1 ; ECONV(1) =>  ENERGY  : EV       -> OUTPUT FORM
C            INTYP = 2 ; ECONV(2) =>  ENERGY  : EV       -> OUTPUT FORM
C            INTYP = 3 ; ECONV(3) =>  ENERGY  : EV/AMU   -> OUTPUT FORM
C AUTHOR:  H. P. SUMMERS, UNIVERSITY OF STRATHCLYDE
C          JA8.08
C          TEL. 0141-553-4196
C DATE:    19/09/95
C UPDATE:  27/08/97  HP SUMMERS - CHANGED NAME FROM CCECON TO CDECON
C MODIFIED: Martin O'Mullane
C DATE:    9-07-98
C VERSION: 1.0 - ported to IDL
C
C VERSION: 1.1 DATE: 01-12-98
C MODIFIED: RICHARD MARTIN
C - PUT UNDER SCCS CONTROL
C
C VERSION:      1.2                                    DATE: 17-05-07
C MODIFIED: Allan Whiteford
C               - Updated comments as part of subroutine documentation
C                 procedure.
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      INTEGER             IEVAL,       INTYP,       OUTTYP
      REAL*8              AMDON,       AMREC,       EIN(IEVAL)
      REAL*8              EOUT(IEVAL)
```

4.67 ceevth: Subroutine ceevth from library adas3xx

```
       SUBROUTINE CEEVTH ( NDENR  ,
     &                     LSETX  , LPASS  ,
     &                     AMDON  , AMREC  , CATYP  , DREN   ,
     &                     ILTYP  , IEXTYP ,
     &                     NENIN  , ENIN   , NENOUT , ENOUT  ,
     &                     SGIN   , RCOUT
     &                   )
C-----------------------------------------------------------------------
c
C  ********************** FORTRAN77 SUBROUTINE: CEEVTH *****************
C
C  VERSION: 1.0
C
C  PURPOSE:  OBTAINS RATE COEFFICIENTS FOR DONOR/RECEIVER CHARGE
C            EXCHANGE COLLISIONS FOR CASES OF
C            MONOENERGETIC DONOR/THERMAL RECEIVER, THERMAL
C            DONOR/MONOENERGETIC RECEIVER, THERMAL DONOR/THERMAL
C            RECEIVER (SAME TEMPERATURE) FROM CROSS-SECTION TABULATIONS.
C
C            A MONO-ENERGETIC CASE IS ALLOWED WHICH CONVERTS INPUT
C            CROSS-SECTIONS TABULATED AT A SET OF ENERGIES/AMU TO
C            OUTPUT CROSS-SECTIONS TABULATED A DIFFERENT SET OF
C            ENERGIES/AMU.
C
C  CALLING PROGRAM:  ADAS314
C
C
C  SUBROUTINE:
C
C  INPUT :  (I*4)  NDENR    = MAX. NUMBER OF ENERGIES/TEMPERATURES
C                             IN INPUT/OUTPUT ENERGY/TEMPERATURE
C                             VECTORS
C  INPUT :  (L*4)  LSETX    = .TRUE. => SPLINE PRESET FOR THESE KNOTS
C                             .FLSE. => SPLINE NOT SET FOR THESE KNOTS
C  INPUT :  (L*4)  LPASS    = .TRUE. => DO NOT CONVERT INTO LOG10 FOR
C                                       ENERGIES AND X-SECTS. FOR SPLINE
C                             .FALSE.=> CONVERT INTO LOG10 FOR
C                                       ENERGIES AND X-SECTS. FOR SPLINE
C  INPUT :  (R*8)  AMDON    = DONOR MASS NUMBER
C  INPUT :  (R*8)  AMREC    = RECEIVER MASS NUMBER
C  INPUT :  (C2)   CATYP    = 'TT' THERMAL/THERMAL (EQUAL TEMPERATURES
C                                  FOR DONOR AND RECEIVER ONLY)
C                             'TR' THERMAL RECEIVER, MONOENERGETIC DONOR
C                             'TD' THERMAL DONOR, MONOENERGETIC RECEIVER
C                             'ME' SPECIAL MONOENERGETIC CASE
C  INPUT : (R*8)   DREN     = DONOR ENERGY    ( 'TR' CASE )
C                             RECEIVER ENERGY ( 'TD' CASE )
C  INPUT : (I*4)   ILTYP    = TYPE FOR LOW AND HIGH ENERGY CROSS-
C                             SECTION EXTRAPOLATION
C                             *** a redundant parameter ***
C                             *** superceeded by IEXTYP ***
C          (I*4)   IEXTYP   = 1 => SET LOWER ENERGIES TO FIRST POINT IN DATA
C                           = 2 => SET LOWER ENERGIES TO 0.0
C  INPUT : (I*4)   NENIN    = NUMBER OF ENERGIES IN INPUT DATA SET
C  INPUT : (R*8)   ENIN()   = ENERGIES (EV/AMU) IN INPUT DATA SET
C  INPUT : (I*4)   NENOUT   = NUMBER OF TEMPERATURES FOR OUTPUT DATA SET
C  INPUT : (R*8)   ENOUT()  = TEMPERATURES (EV) FOR OUTPUT DATA SET FOR
C                             'TT', 'TD', 'TR' CASES.
C                           = ENERGY/AMU FOR OUTPUT DATA SET FOR
```

```
C                               'ME' CASE.
C  INPUT : (R*8)   SGIN()    = INPUT X-SECTIONS (CM2) FROM INPUT DATA SET
C                               1ST.DIM: ENERGY INDEX
C  OUTPUT: (R*8)   RCOUT()   = RATE COEFF. (CM3 S-1) IN OUTPUT DATA SET
C                               1ST.DIM: TEMPERATURE INDEX
C          (I*4)   I         = GENERAL INDEX
C          (I*4)   IT        = GENERAL INDEX
C          (I*4)   ITHETA    = GENERAL INDEX
C          (I*4)   IOPT      = SPLINE END POINT CURVATURE/GRADIENT OPTION
C                               1 => DDY1 = 0, DDYN = 0
C                               4 => DY1 = 0 , DDYN = 0
C          (I*4)   IXD       = DONOR GAUSSIAN QUADRATURE INDEX
C          (I*4)   IXR       = RECEIVER GAUSSIAN QUADRATURE INDEX
C          (I*4)   NGS       = GAUSSIAN QUARATURE DIMENSION
C          (I*4)   NTHETA    = NUMBER OF ANGLE VALUES FOR QUADRATURE
C          (I*4)   LTHETA    = NTHETA+1
C          (I*4)   L1        = PARAMETER = 1
C          (R*8)   ETHD      = THERMAL ENERGY OF DONOR       (JOULES)
C          (R*8)   ETHR      = THERMAL ENERGY RECEIVER       (JOULES)
C          (R*8)   HSIMP     = SIMPSON'S RULE STEP INTERVAL
C          (R*8)   THETA     = ANGLE BETWEEN PARTICLE VELOCITIES (RAD)
C          (R*8)   FAC       = GENERAL VARIABLE
C          (R*8)   FLAG      = GENERAL VARIABLE
C          (R*8)   XMDKG     = DONOR MASS    (KG)
C          (R*8)   XMRKG     = RECEIVER MASS (KG)
C          (R*8)   VD        = DONOR SPEED    (M S-1)
C          (R*8)   VR        = RECEIVER SPEED (M S-1)
C          (R*8)   RATE      = EVALUATED RATE COEFFICIENT (CM3 S-1)
C          (R*8)   PART1     = GENERAL VARIABLE
C          (R*8)   PART2     = GENERAL VARIABLE
C          (R*8)   PART3     = GENERAL VARIABLE
C          (R*8)   PART12    = GENERAL VARIABLE
C          (R*8)   PART23    = GENERAL VARIABLE
C          (R*8)   PART123   = GENERAL VARIABLE
C          (R*8)   VREL1     = GENERAL RELATIVE SPEED VARIABLE
C          (R*8)   XSEC1     = GENERAL CROSS-SECTION VARIABLE
C          (R*8)   VAL       = GENERAL VARIABLE
C          (R*8)   XGS()     = GAUSSIAN QUADRATURE NODES
C          (R*8)   WGS()     = GAUSSIAN QUADRATURE WEIGHTS
C          (R*8)   VREL()    = RELATIVE SPEED OF PARTICLES FOR DIFFERENT
C                               ANGLES (CM S-1)
C          (R*8)   XSEC()    = CHARGE EXCHANGE CROSS-SECTIONS FOR
C                               RELATIVE SPEEDS AT DIFFERENT ANGLES (CM2)
C
C
C  ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          CESGCX     ADAS      INTERPOLATES CX CROSS-SECTION TABLES
C
C
C  AUTHOR:  H. P. SUMMERS, UNIVERSITY OF STRATHCLYDE
C           JA8.08
C           TEL. 0141-553-4196
C  DATE:    02/11/95
C
C  UPDATE:  09/07/98  Martin O'Mullane
C           CHANGED NAME FROM CXTHER TO CDEVTH. SIMILAR FUNCTIONALITY
C           BUT IS EXTENDED TO DEAL WITH EXTRA AVERAGING METHODS.
C
C VERSION: 1.1 DATE: 01-12-98
```

```
C MODIFIED: RICHARD MARTIN
C - PUT UNDER SCCS CONTROL
C
C VERSION: 1.2 DATE: 24-03-99
C MODIFIED: MARTIN O'MULLANE
C - SECOND VERSIONS
C
C VERSION:      1.3                                     DATE: 17-05-07
C MODIFIED: Allan Whiteford
C               - Updated comments as part of subroutine documentation
C                 procedure.
C-----------------------------------------------------------------------
      CHARACTER*2         CATYP
      INTEGER             IEXTYP,      ILTYP,       NDENR,       NENIN
      INTEGER             NENOUT
      LOGICAL             LPASS,       LSETX
      REAL*8              AMDON,       AMREC,       DREN
      REAL*8              ENIN(NDENR), ENOUT(NDENR)
      REAL*8              RCOUT(NDENR),             SGIN(NDENR)
```

4.68 cefill: Subroutine cefill from library adas3xx

```
       SUBROUTINE CEFILL( MXNENG , MXNSHL ,
     &                    SYMBR  , SYMBD  , IZR    , IZD    ,
     &                    INDD   , NENRGY , NMIN   , NMAX   ,
     &                    LPARMS , LSETL  , ENRGYA ,
     &                    ALPHAA , LFORMA , XLCUTA , PL2A   ,
     &                    PL3A   , SIGTA  , SIGNA  , SIGLA
     &                  )
C-----------------------------------------------------------------------
C  ****************** FORTRAN77 SUBROUTINE: CDFILL *********************
C
C  PURPOSE:  FILL HIGH N ZEROES IN AN ADF01 IF PRESENT.
C
C  CALLING PROGRAM: ADAS314
C  DATA:
C           THE UNITS USED IN THE DATA FILE ARE TAKEN AS FOLLOWS:
C           COLLISION ENERGIES  : KEV/AMU
C           ALPHA               :
C           TOTAL XSECTS.       : CM2
C           N-SHELL XSECTS.     : CM2
C           NL-SHELL DATA       : CM2
C           NLM-SHELL DATA      : CM2
C  SUBROUTINE:
C  INPUT : (I*4)  MXNENG    = MAXIMUM NO. OF ENERGIES.
C  INPUT : (I*4)  MXNSHL    = MAXIMUM NO. OF N SHELLS.
C  INPUT : (C*2)  SYMBR     = READ - RECEIVER ION ELEMENT SYMBOL.
C  INPUT : (C*2)  SYMBD     = READ - DONOR ION ELMENT SYMBOL.
C  INPUT : (I*4)  IZR       = READ - ION CHARGE OF RECEIVER.
C  INPUT : (I*4)  IZD       = READ - ION CHARGE OF DONOR.
C  INPUT : (I*4)  INDD      = READ - DONOR STATE INDEX.
C  INPUT : (I*4)  NENRGY    = NUMBER OF ENERGIES READ.
C  INPUT : (I*4)  NMIN      = LOWEST N-SHELL FOR WHICH DATA READ.
C  INPUT : (I*4)  NMAX      = HIGHEST N-SHELL FOR WHICH DATA READ.
C  INPUT : (L*4)  LPARMS    = FLAGS IF L-SPLITTING PARAMETERS PRESENT.
C                             .TRUE.  => L-SPLITTING PARAMETERS PRESENT.
C                             .FALSE  => L-SPLITTING PARAMETERS ABSENT.
C  INPUT : (L*4)  LSETL     = FLAGS IF L-RESOLVED DATA PRESENT.
C                             .TRUE.  => L-RESOLVED DATA PRESENT.
C                             .FALSE  => L-RESOLVED DATA ABSENT.
C  INPUT : (R*8)  ENRGYA()  = READ - COLLISION ENERGIES.
C                             UNITS: EV/AMU (READ AS KEV/AMU)
C                             DIMENSION: ENERGY INDEX
C  INPUT : (R*8)  ALPHAA()  = READ - EXTRAPOLATION PARAMETER ALPHA.
C                             DIMENSION: ENERGY INDEX
C  INPUT : (I*4)  LFORMA()  = READ - PARAMETERS FOR CALCULATING L-RES
C                                    X-SEC.
C                             DIMENSION: ENERGY INDEX
C  INPUT : (R*8)  XLCUTA()  = READ - PARAMETERS FOR CALCULATING L-RES
C                                    X-SEC.
C                             DIMENSION: ENERGY INDEX
C  INPUT : (R*8)  PL2A()    = READ - PARAMETERS FOR CALCULATING L-RES
C                                    X-SEC.
C                             DIMENSION: ENERGY INDEX
C  INPUT : (R*8)  PL3A()    = READ - PARAMETERS FOR CALCULATING L-RES
C                                    X-SEC.
C                             DIMENSION: ENERGY INDEX
C  I/O   : (R*8)  SIGTA()   = READ - TOTAL CHARGE EXCHANGE
C                                    CROSS-SECTION.
C                             UNITS: CM2
C                             DIMENSION: ENERGY INDEX
```

```
C  I/O   : (R*8)  SIGNA(,)  = READ - N-RESOLVED CHARGE EXCHANGE
C                                  CROSS-SECTIONS.
C                              UNITS: CM2
C                              1ST DIMENSION: ENERGY INDEX
C                              2ND DIMENSION: N-SHELL
C  I/O   : (R*8)  SIGLA(,)  = READ - L-RESOLVED CHARGE EXCHANGE
C                                  CROSS-SECTIONS.
C                              UNITS: CM2
C                              1ST DIMENSION: ENERGY INDEX
C                              2ND DIMENSION: INDEXED BY I4IDFL(N,L)
C  I/O   : (R*8)  SIGMA(,)  = READ - M-RESOLVED CHARGE EXCHANGE
C                                  CROSS-SECTIONS.
C                              UNITS: CM2
C                              1ST DIMENSION: ENERGY INDEX
C                              2ND DIMENSION: INDEXED BY I4IDFM(N,L,M)
C                                             WITH M >= 0 ONLY
C          (I*4)  I        = N QUANTUM NUMBER.
C          (I*4)  J        = L QUANTUM NUMBER.
C          (I*4)  K        = M QUANTUM NUMBER.
C          (I*4)  N        = N QUANTUM NUMBER.
C          (I*4)  L1       = L QUANTUM NUMBER + 1
C          (I*4)  M1       = M QUANTUUM NUMBER + 1
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4IDFL     ADAS      RETURNS UNIQUE INDEX FROM QUANTUM
C                               NUMBERS N AND L.
C          I4IDFM     ADAS      RETURNS UNIQUE INDEX FROM QUANTUM
C                               NUMBERS N, L AND M.
C AUTHOR:  H. P. SUMMERS, UNIVERSITY OF STRATHCLYDE
C          JA8.08
C          TEL 0141-553-4196
C DATE:    21/09/95
C UPDATE:  27/08/97  HP SUMMERS - CHANGED NAME FROM CCFILL TO CDFILL
C
C VERSION: 1.1 DATE: 01-12-98
C MODIFIED: RICHARD MARTIN
C - PUT UNDER SCCS CONTROL
C
C VERSION:      1.2                                    DATE: 17-05-07
C MODIFIED: Allan Whiteford
C               - Updated comments as part of subroutine documentation
C                 procedure.
C
C VERSION  : 1.3
C DATE     : 22-05-2007
C MODIFIED : Martin O'Mullane
C              - Remove unused m-subshell data possibility.
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      CHARACTER*2         SYMBD,       SYMBR
      INTEGER             INDD,        IZD,         IZR
      INTEGER             LFORMA(MXNENG),           MXNENG,      MXNSHL
      INTEGER             NENRGY,      NMAX,        NMIN
      LOGICAL             LPARMS,      LSETL
      REAL*8              ALPHAA(MXNENG),           ENRGYA(MXNENG)
      REAL*8              PL2A(MXNENG),             PL3A(MXNENG)
      REAL*8              SIGLA(MXNENG,(MXNSHL*(MXNSHL+1))/2)
      REAL*8              SIGNA(MXNENG,MXNSHL),     SIGTA(MXNENG)
      REAL*8              XLCUTA(MXNENG)
```

4.69 ceparm: Subroutine ceparm from library adas3xx

```
       SUBROUTINE CEPARM ( NDENR  ,
     &                     LPARMS ,
     &                     NENIN  , ENIN   , NENOUT , ENOUT  ,
     &                     ALFIN  , XLCIN  , PL2IN  , PL3IN  , LFMIN  ,
     &                     ALFOUT , XLCOUT , PL2OUT , PL3OUT , LFMOUT
     &                   )
C-----------------------------------------------------------------------
C  ********************** FORTRAN77 SUBROUTINE: CEPARM *****************
C  VERSION: 1.0
C
C  PURPOSE:  CONVERTS ALPHA, PL2, PL3 AND LFORM CHARGE EXCHANGE
C            PARAMETER VALUES AT INPUT ENERGIES TO VALUES AT OUTPUT
C            ENERGIES
C
C  CALLING PROGRAM:  ADAS314
C  SUBROUTINE:
C  INPUT :  (I*4)  NDENR   = MAX. NUMBER OF ENERGIES
C                            ALLOWED IN CROSS-SECTION FILE
C                            OR TEMPERATURES IN THERMAL
C                            AVERAGED RATE COEFFT. OUTPUT FILE.
C  INPUT : (I*4)   LPARMS   = .TRUE.  => INPUT DATA HAS L-FIT PARAMETERS
C                             .FALSE. => INPUT DATA HAS L-FIT PARAMETERS
C  INPUT : (I*4)   NENIN    = NUMBER OF ENERGIES IN INPUT DATA SET
C  INPUT : (R*8)   ENIN()   = ENERGIES (EV/AMU) IN INPUT DATA SET
C  INPUT : (I*4)   NENOUT   = NUMBER OF ENERGIES FOR OUTPUT DATA SET
C  INPUT : (R*8)   ENOUT()  = TEMPERATURES (EV/AMU) FOR OUTPUT DATA SET
C  INPUT : (R*8)   ALFIN()  = ALPHA PARAMETER IN INPUT DATA SET
C                             1ST.DIM: ENERGY INDEX
C  INPUT : (R*8)   XLCIN()  = NON-INTEGER L-CUT-OFF PARAMETER
C                             1ST.DIM: ENERGY INDEX
C  INPUT : (R*8)   PL2IN(,) = P2 L-FIT PARAMETER IN INPUT DATA SET
C                             1ST.DIM: ENERGY INDEX
C  INPUT : (R*8)   PL3IN(,) = P3 L-FIT PARAMETER IN INPUT DATA SET
C                             1ST.DIM: ENERGY INDEX
C  INPUT : (I*4)   LFMIN(,) = L-FIT FORM TYPE INDEX IN INPUT DATA SET
C                             1ST.DIM: ENERGY INDEX
C  OUTPUT: (R*8)   ALFOUT() = ALPHA PARAMETER IN INPUT DATA SET
C                             1ST.DIM: ENERGY INDEX
C  OUTPUT: (R*8)   XLCOUT() = NON-INTEGER L-CUT-OFF PARAMETER
C                             1ST.DIM: ENERGY INDEX
C  OUTPUT: (R*8)   PL2OUT() = P2 L-FIT PARAMETER IN INPUT DATA SET
C                             1ST.DIM: ENERGY INDEX
C  OUTPUT: (R*8)   PL3OUT() = P3 L-FIT PARAMETER IN INPUT DATA SET
C                             1ST.DIM: ENERGY INDEX
C  OUTPUT: (I*4)   LFMOUT() = L-FIT FORM TYPE INDEX IN INPUT DATA SET
C                             1ST.DIM: ENERGY INDEX
C          (L*4)   LSETX    = .TRUE. => SPLINE PRESET FOR THESE KNOTS
C                             .FLSE. => SPLINE NOT SET FOR THESE KNOTS
C          (L*4)   LPASS    = .TRUE. => DO NOT CONVERT INTO LOG10 FOR
C                                       ENERGIES AND X-SECTS. FOR SPLINE
C                             .FLSE. => CONVERT INTO LOG10 FOR
C                                       ENERGIES AND X-SECTS. FOR SPLINE
C          (I*4)   IOPT     = SPLINE END POINT CURVATURE/GRADIENT OPTION
C                             1 => DDY1 = 0, DDYN = 0
C                             4 => DY1 = 0 , DDYN = 0
C          (R*8)   CMSAMU   = PARAMETER = CONVERSION FACTOR FOR ENERGY
C                                         (AMU) TO VELOCITY (CM S-1)
C          (I*4)   I        = GENERAL INDEX
C          (I*4)   IT       = GENERAL INDEX
```

```
C          (R*8)   XIN()    = INTERNAL SPLINE INDEPENDENT VARIABLE
C          (R*8)   YIN()    = INTERNAL SPLINE DEPENDENT VARIABLE
C          (R*8)   VIN()    = INTERNAL VECTOR
C          (R*8)   DY()     = DERIVATIVES AT SPLINE KNOTS
C          (R*8)   XOUT()   = INTERNAL OUTPUT INDEPENDENT VARIABLE
C          (R*8)   YOUT()   = INTERNAL OUTPUT DEPENDENT VARIABLE
C          (L*4)   LINTRP() = .TRUE.  => POINT INTERPOLATED
C                           = .FALSE. => POINT EXTRAPOLATED
C  ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          XXSPLE     ADAS      INTERPOLATES USING CUBIC SPLINES
C          R8FUN1      ADAS      EXTERNAL FUNCTION FOR XXSPLE
C  AUTHOR:  H. P. SUMMERS, UNIVERSITY OF STRATHCLYDE
C           JA8.08
C           TEL. 0141-553-4196
C  DATE:    13/11/95
C  UPDATE:  27/08/97  HP SUMMERS - CHANGED NAME FROM CCPARM TO CDPARM
C
C VERSION: 1.1 DATE: 01-12-98
C MODIFIED: RICHARD MARTIN
C - PUT UNDER SCCS CONTROL
C
C VERSION:      1.2                                     DATE: 17-05-07
C MODIFIED: Allan Whiteford
C               - Updated comments as part of subroutine documentation
C                 procedure.
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      INTEGER             LFMIN(NDENR),             LFMOUT(NDENR)
      INTEGER             NDENR,       NENIN,       NENOUT
      LOGICAL             LPARMS
      REAL*8              ALFIN(NDENR),             ALFOUT(NDENR)
      REAL*8              ENIN(NDENR), ENOUT(NDENR)
      REAL*8              PL2IN(NDENR),             PL2OUT(NDENR)
      REAL*8              PL3IN(NDENR),             PL3OUT(NDENR)
      REAL*8              XLCIN(NDENR),             XLCOUT(NDENR)
```

4.70 cesgcx: Subroutine cesgcx from library adas3xx

```
      SUBROUTINE CESGCX ( LSETX    , LPASS    , ILTYP    , IOPT     ,
     &                    NENIN    , ENIN     , SGIN     ,
     &                    LTHETA   , VREL     , XSEC
     &                  )
C-----------------------------------------------------------------------
C
C  ********************** FORTRAN77 SUBROUTINE: CESGCX *****************
C
C  VERSION: 1.0 (ADAS91)
C
C  PURPOSE:  INTERPOLATES CROSS-SECTION DATA FROM AN INPUT VECTOR OF
C            VALUES USING CUBIC SPLINES.
C
C            EXTRAPOLATES FOR RELATIVE SPEEDS OUT OF DATA RANGE
C            ACCORDING TO VARIOUS TYPES (ILTYP).  LOGARITHMIC
C            INTERPOLATION MAY BE USED (LPASS). SPEED ECONOMY IS
C            POSSIBLE FOR REPEATS WITH THE SAME SPLINE KNOTS (LSETX).
C
C  CALLING PROGRAM:  CEEVTH
C
C  NOTES:
C        (1) FOR  ILTYP.EQ.0, EXTRAPOLATION IS AS FOLLOWS:
C              XSEC = SIG0*DEXP(-ALPH0/VREL) FOR VREL<VREL(MIN)
C              XSEC = SIG1*VREL**(-7.0)  FOR VREL> VREL(MAX),
C              WHERE VREL(MIN), VREL(MAX) ARE THE FIRST AND LAST FROM
C              INPUT VALUES IN DATA TABLES IN ADF24.
C            FOR  ILTYP.NE.0, EXTRAPOLATION IS AS AS ABOVE AT THIS
C              TIME.
C
C        (2) Based on sigcx.for but is a special version for ADAS314.
C
C  SUBROUTINE:
C
C  INPUT : (L*4)   LSETX    = .TRUE. => SPLINE NOT SET FOR THESE KNOTS
C                             .FLSE. => SPLINE NOT FOR THESE KNOTS
C  INPUT : (L*4)   LPASS    = .TRUE. => DO NOT CONVERT INTO LOG10 FOR
C                                       ENERGIES AND X-SECTS. FOR SPLINE
C                             .FLSE. => CONVERT INTO LOG10 FOR
C                                       ENERGIES AND X-SECTS. FOR SPLINE
C  INPUT : (I*4)   ILTYP    = TYPE FOR LOW AND HIGH ENERGY CROSS-
C                             -SECTION EXTRAPOLATION.
C  INPUT : (I*4)   IOPT     = SPLINE END POINT CURVATURE/GRADIENT OPTION
C                             1 => DDY1 = 0, DDYN = 0
C                             4 => DY1 = 0 , DDYN = 0
C
C  INPUT : (I*4)   NENIN    = NUMBER OF ENERGIES IN INPUT DATA SET
C  INPUT : (R*8)   ENIN()   = ENERGIES (EV/AMU) IN INPUT DATA SET
C  INPUT : (R*8)   SGIN()   = INPUT X-SECTIONS (CM2) FROM INPUT DATA SET
C                             1ST.DIM: ENERGY INDEX
C  INPUT : (I*4)   LTHETA   = NUMBER OF VALUES IN VREL VECTOR
C  INPUT : (R*8)   VREL()   = RELATIVE SPEEDS FOR OUTPUT (CM S-1)
C
C  OUTPUT: (R*8)   XSEC()   = OUTPUT CROSS-SECTION (CM2)
C
C          (I*4)   MAXENS   = PARAMETER = MAX. LENGTH OF TABULAR XSECT.
C                                         VECTOR
C          (I*4)   LDTHET   = PARAMETER = MAX. LENGTH OF INTERNAL
C                                         VECTORS
C          (R*8)   CMSAMU   = PARAMETER = CONVERSION FACTOR FOR ENERGY
```

```
C                                        (AMU) TO VELOCITY (CM S-1)
C
C          (I*4)   I        = GENERAL INDEX
C          (I*4)   N        = GENERAL INDEX
C          (R*8)   ALPH0    = LOW VELOCITY EXTRAPOLATION PARAMETER
C          (R*8)   EXPON    = EXPONENT OF EXPONENTIAL
C          (R*8)   VSLOPE   = HIGH VELOCITY EXTRAPOLATION PARAMETER
C          (R*8)   XIN()    = INTERNAL SPLINE INDEPENDENT VARIABLE
C          (R*8)   YIN()    = INTERNAL SPLINE DEPENDENT VARIABLE
C          (R*8)   VIN()    = INTERNAL VECTOR
C          (R*8)   DY()     = DERIVATIVES AT SPLINE KNOTS
C          (R*8)   XOUT()   = INTERNAL OUTPUT INDEPENDENT VARIABLE
C          (R*8)   YOUT()   = INTERNAL OUTPUT DEPENDENT VARIABLE
C          (L*4)   LINTRP() = .TRUE.  => POINT INTERPOLATED
C                           = .FALSE. => POINT EXTRAPOLATED
C
C
C  ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          XXSPLE     ADAS      INTERPOLATES USING CUBIC SPLINES
C          R8FUN1     ADAS      EXTERNAL FUNCTION FOR XXSPLE
C
C
C  AUTHOR:  H. P. SUMMERS, UNIVERSITY OF STRATHCLYDE
C           JA8.08
C           TEL. 0141-553-4196
C
C  DATE:    03/11/95
C
C  UPDATE:  11/04/96  HP SUMMERS - TRAPPED CASE OF ZERO RELATIVE SPEED
C
C UNIX-IDL PORT: H.P.SUMMERS
C
C VERSION: 1.1 DATE: 18-02-99
C MODIFIED: Martin O'Mullane
C           - Based on SIGCX.FOR
C
C VERSION: 1.2 DATE: 17-05-07
C MODIFIED: Allan Whiteford
C               - Updated comments as part of subroutine documentation
C                 procedure.
C
C-----------------------------------------------------------------------
      INTEGER             ILTYP,       IOPT,        LTHETA,      NENIN
      LOGICAL             LPASS,       LSETX
      REAL*8              ENIN(NENIN), SGIN(NENIN), VREL(LTHETA)
      REAL*8              XSEC(LTHETA)
```

4.71 cether: Subroutine cether from library adas3xx

```
       SUBROUTINE CETHER ( NDENR  , NDSHL  ,
     &                     AMDON  , AMREC  , CATYP  , DREN   ,
     &                     ILTYP  , IEXTYP ,
     &                     LSETL  , NMIN   , NMAX   ,
     &                     NENIN  , ENIN   , NENOUT , ENOUT  ,
     &                     SGTIN  , SGNIN  , SGLIN  ,
     &                     RCTOUT , RCNOUT , RCLOUT
     &                   )
C-----------------------------------------------------------------------
C
C  ********************** FORTRAN77 SUBROUTINE: CETHER *****************
C
C  VERSION: 1.0
C
C  PURPOSE:  OBTAINS RATE COEFFICIENTS FOR DONOR/RECEIVER CHARGE
C            EXCHANGE COLLISIONS FOR CASES OF
C            MONOENERGETIC DONOR/THERMAL RECEIVER, THERMAL
C            DONOR/MONOENERGETIC RECEIVER, THERMAL DONOR/THERMAL
C            RECEIVER (SAME TEMPERATURE) FROM CROSS-SECTION TABULATIONS
C
C  CALLING PROGRAM:  ADAS314
C
C  SUBROUTINE:
C  INPUT :  (I*4)  NDENR   = MAX. NUMBER OF ENERGIES
C                            ALLOWED IN CROSS-SECTION FILE
C                            OR TEMPERATURES IN THERMAL
C                            AVERAGED RATE COEFFT. OUTPUT FILE.
C  INPUT :  (I*4)  NDSHL   = PARAMETER = MAX. NUMBER OF N-SHELLS
C                                        ALLOWED IN CROSS-SECTION FILE
C  INPUT :  (R*8)  AMDON    = DONOR MASS NUMBER
C  INPUT :  (R*8)  AMREC    = RECEIVER MASS NUMBER
C  INPUT :  (C2)   CATYP    = 'TT' THERMAL/THERMAL (EQUAL TEMP. CASE)
C                              'TR' THERMAL RECEIVER, MONOENERGETIC DONOR
C                              'TD' THERMAL DONOR, MONOENERGETIC RECEIVER
C  INPUT : (R*8)   DREN     = DONOR ENERGY    ( 'TR' CASE )
C                              RECEIVER ENERGY ( 'TD' CASE )
C  INPUT : (I*4)   ILTYP    = TYPE FOR LOW ENERGY CROSS-SECTION EXTRAPOL
C                              REDUNDANT - SUPERCEEDED BY IEXTYP
C          (I*4)   IEXTYP   = 1 => SET LOWER ENERGIES TO FIRST POINT IN DATA
C                           = 2 => SET LOWER ENERGIES TO 0.0
C  INPUT : (L*4)   LSETL    = .TRUE.  => L-RESOLVED DATA READ
C                              .FALSE. => NO L-RESOLVED DATA READ
C  INPUT : (L*4)   LSETM    = .TRUE.  => M-RESOLVED DATA READ
C                              .FALSE. => NO M-RESOLVED DATA READ
C  INPUT : (I*4)   NMIN     = LOWEST N-SHELL IN DATA SET
C  INPUT : (I*4)   NMAX     = HIGHEST N-SHELL IN DATA SET
C  INPUT : (I*4)   NENIN    = NUMBER OF ENERGIES IN INPUT DATA SET
C  INPUT : (R*8)   ENIN()   = ENERGIES (EV/AMU) IN INPUT DATA SET
C  INPUT : (I*4)   NENOUT   = NUMBER OF ENERGIES FOR OUTPUT DATA SET
C  INPUT : (R*8)   ENOUT()  = TEMPERATURES (EV/AMU) FOR OUTPUT DATA SET
C  INPUT : (R*8)   SGTIN()  = TOTAL X-SECTIONS (CM2) IN INPUT DATA SET
C                              1ST.DIM: ENERGY INDEX
C  INPUT : (R*8)   SGNIN(,) = N-SHELL X-SECTIONS (CM2) IN INPUT DATA SET
C                              1ST.DIM: ENERGY INDEX
C                              2ND.DIM: PRINCIPAL QUANTUM NUMBER
C  INPUT : (R*8)   SGLIN(,) = L-SHELL X-SECTIONS (CM2) IN INPUT DATA SET
C                              1ST.DIM: ENERGY INDEX
C                              2ND.DIM: NL REFERENCE INDEX
C  INPUT : (R*8)   SGMIN(,) = M-SHELL X-SECTIONS (CM2) IN INPUT DATA SET
```

```
C                              1ST.DIM: ENERGY INDEX
C                              2ND.DIM: NLM REFERENCE INDEX
C  OUTPUT: (R*8)   RCTOUT() = TOTAL RATE COEFFT. (CM3 S-1) FOR OUTPUT
C                              1ST.DIM: TEMPERATURE INDEX
C  OUTPUT: (R*8)   RCNOUT(,)= N-SHELL RATE COEFFT. (CM3 S-1) FOR OUTPUT
C                              1ST.DIM: TEMPERATURE INDEX
C                              2ND.DIM: PRINCIPAL QUANTUM NUMBER
C  OUTPUT: (R*8)   RCLOUT(,)= L-SHELL RATE COEFFT. (CM3 S-1) FOR OUTPUT
C                              1ST.DIM: TEMPERATURE INDEX
C                              2ND.DIM: NL REFERENCE INDEX
C  OUTPUT: (R*8)   RCMOUT(,)= M-SHELL RATE COEFFT. (CM3 S-1) FOR OUTPUT
C                              1ST.DIM: TEMPERATURE INDEX
C                              2ND.DIM: NLM REFERENCE INDEX
C                  LSETX    = .TRUE. => SPLINE PRESET FOR THESE KNOTS
C                             .FLSE. => SPLINE NOT SET FOR THESE KNOTS
C                  LPASS    = .TRUE. => DO NOT CONVERT INTO LOG10 FOR
C                                       ENERGIES AND X-SECTS. FOR SPLINE
C                             .FLSE. => CONVERT INTO LOG10 FOR
C                                       ENERGIES AND X-SECTS. FOR SPLINE
C          (I*4)   PIPEOU   = STANDARD OUTPUT
C
C
C  ROUTINES:
C
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          CDEVTH     ADAS      EVALUATES THERMAL AVERAGE RATE COEFFTS.
C          I4FCTN     ADAS      RETURNS CHARACTER STRING AS AN INTEGER.
C          I4UNIT     ADAS      FETCH UNIT NUMBER FOR OUTPUT OF MESSAGES
C          I4IDFL     ADAS      RETURNS UNIQUE INDEX FROM QUANTUM
C                               NUMBERS N AND L.
C          I4IDFM     ADAS      RETURNS UNIQUE INDEX FROM QUANTUM
C                               NUMBERS N, L AND M.
C
C
C  AUTHOR:  H. P. SUMMERS, UNIVERSITY OF STRATHCLYDE
C           JA8.08
C           TEL. 0141-553-4196
C  DATE:    10/10/95
C  UPDATE:  27/08/97  HP SUMMERS - CHANGED NAME FROM CCTHER TO CDTHER
C
C VERSION: 1.1 DATE: 01-12-98
C MODIFIED: RICHARD MARTIN
C - PUT UNDER SCCS CONTROL
C
C VERSION:      1.2                                     DATE: 24-03-99
C MODIFIED: MARTIN O'MULLANE
C               - SECOND VERSION
C
C VERSION  : 1.3
C DATE     : 22-05-2007
C MODIFIED : Martin O'Mullane
C              - Remove unused m-subshell data possibility.
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      CHARACTER*2         CATYP
      INTEGER             IEXTYP,      ILTYP,       NDENR,       NDSHL
      INTEGER             NENIN,       NENOUT,      NMAX,        NMIN
      LOGICAL             LSETL
      REAL*8              AMDON,       AMREC,       DREN
```

```
REAL*8              ENIN(NDENR), ENOUT(NDENR)
REAL*8              RCLOUT(NDENR,(NDSHL*(NDSHL+1))/2)
REAL*8              RCNOUT(NDENR,NDSHL),      RCTOUT(NDENR)
REAL*8              SGLIN(NDENR,(NDSHL*(NDSHL+1))/2)
REAL*8              SGNIN(NDENR,NDSHL),       SGTIN(NDENR)
```

4.72 cewr11: Subroutine cewr11 from library adas3xx

```
       SUBROUTINE CEWR11( DSFULL , DATE   ,
     &                    IUNIT  , MXNENG , MXNSHL ,
     &                    CATYP  , AMDON  , AMREC  , DREN   ,
     &                    SYMBR  , SYMBD  , IZR    , IZD    ,
     &                    INDD   , NENRGY , NMIN   , NMAX   ,
     &                    LPARMS , LSETL  , ENRGYA ,
     &                    ALPHAA , LFORMA , XLCUTA , PL2A   ,
     &                    PL3A   , SIGTA  , SIGNA  , SIGLA
     &                  )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: CEWR11 *********************
C
C  PURPOSE:  TO OUTPUT DATA TO MODIFIED ADF01 FILE.
C
C            IF THE ADF01 FILE IS FOR THERMAL/THERMAL,
C            THERMAL/DONOR OR THERMAL/RECEIVER THEN THE RATE
C            (PASSED AS SIGMA) IS DIVIDED BY
C                   1.384D4 * 100 * DSQRT(TE)
C            IN ORDER TO ALLOW THE FILE TO BE USED WITH UNMODIFIED
C            SERIES 3 PROGRAMS.
C
C  CALLING PROGRAM: ADAS314
C
C  DATA:
C           THE UNITS USED IN THE DATA FILE ARE TAKEN AS FOLLOWS:
C           COLLISION ENERGIES  : KEV/AMU
C           ALPHA               :
C           TOTAL XSECTS.       : CM2
C           N-SHELL XSECTS.     : CM2
C           NL-SHELL DATA       : CM2
C           NLM-SHELL DATA      : CM2
C
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  IUNIT     = INPUT  UNIT NUMBER FOR RESULTS
C  INPUT : (I*4)  MXNENG    = MAXIMUM NO. OF ENERGIES.
C  INPUT : (I*4)  MXNSHL    = MAXIMUM NO. OF N SHELLS.
C  INPUT : (C*80) TITLED    = NOT SET - TITLE FOR DATA SOURCE.
C  INPUT : (C*80) DSFULL    = SOURCE DATASET
C  INPUT : (C2)   CATYP     = 'TT' THERMAL/THERMAL (EQUAL TEMPERATURES
C                                FOR DONOR AND RECEIVER ONLY)
C                            'TR' THERMAL RECEIVER, MONOENERGETIC DONOR
C                            'TD' THERMAL DONOR, MONOENERGETIC RECEIVER
C                            'ME' SPECIAL MONOENERGETIC CASE
C  INPUT : (R*8)  AMDON     = DONOR MASS NUMBER
C  INPUT : (R*8)  AMREC     = RECEIVER MASS NUMBER
C  INPUT : (R*8)  DREN      = DONOR ENERGY    ( 'TR' CASE )
C                             RECEIVER ENERGY ( 'TD' CASE )
C  INPUT : (C*2)  SYMBR     = READ - RECEIVER ION ELEMENT SYMBOL.
C  INPUT : (C*2)  SYMBD     = READ - DONOR ION ELMENT SYMBOL.
C  INPUT : (I*4)  IZR       = READ - ION CHARGE OF RECEIVER.
C  INPUT : (I*4)  IZD       = READ - ION CHARGE OF DONOR.
C  INPUT : (I*4)  INDD      = READ - DONOR STATE INDEX.
C  INPUT : (I*4)  NENRGY    = NUMBER OF ENERGIES READ.
C  INPUT : (I*4)  NMIN      = LOWEST N-SHELL FOR WHICH DATA READ.
C  INPUT : (I*4)  NMAX      = HIGHEST N-SHELL FOR WHICH DATA READ.
C  INPUT : (L*4)  LPARMS    = FLAGS IF L-SPLITTING PARAMETERS PRESENT.
```

```
C                               .TRUE.  => L-SPLITTING PARAMETERS PRESENT.
C                               .FALSE  => L-SPLITTING PARAMETERS ABSENT.
C  INPUT : (L*4)  LSETL     = FLAGS IF L-RESOLVED DATA PRESENT.
C                               .TRUE.  => L-RESOLVED DATA PRESENT.
C                               .FALSE  => L-RESOLVED DATA ABSENT.
C  INPUT : (L*4)  LSETM     = FLAGS IF M-RESOLVED DATA PRESENT.
C                               .TRUE.  => M-RESOLVED DATA PRESENT.
C                               .FALSE  => M-RESOLVED DATA ABSENT.
C  INPUT : (R*8)  ENRGYA()  = READ - COLLISION ENERGIES.
C                               UNITS: EV/AMU (READ AS KEV/AMU)
C                               DIMENSION: ENERGY INDEX
C  INPUT : (R*8)  ALPHAA()  = READ - EXTRAPOLATION PARAMETER ALPHA.
C                               DIMENSION: ENERGY INDEX
C  INPUT : (I*4)  LFORMA()  = READ - PARAMETERS FOR CALCULATING L-RES
C                                     X-SEC.
C                               DIMENSION: ENERGY INDEX
C  INPUT : (R*8)  XLCUTA()  = READ - PARAMETERS FOR CALCULATING L-RES
C                                     X-SEC.
C                               DIMENSION: ENERGY INDEX
C  INPUT : (R*8)  PL2A()    = READ - PARAMETERS FOR CALCULATING L-RES
C                                     X-SEC.
C                               DIMENSION: ENERGY INDEX
C  INPUT : (R*8)  PL3A()    = READ - PARAMETERS FOR CALCULATING L-RES
C                                     X-SEC.
C                               DIMENSION: ENERGY INDEX
C  INPUT : (R*8)  SIGTA()   = READ - TOTAL CHARGE EXCHANGE
C                                     CROSS-SECTION.
C                               UNITS: CM2
C                               DIMENSION: ENERGY INDEX
C  INPUT : (R*8)  SIGNA(,)  = READ - N-RESOLVED CHARGE EXCHANGE
C                                     CROSS-SECTIONS.
C                               UNITS: CM2
C                               1ST DIMENSION: ENERGY INDEX
C                               2ND DIMENSION: N-SHELL
C  INPUT : (R*8)  SIGLA(,)  = READ - L-RESOLVED CHARGE EXCHANGE
C                                     CROSS-SECTIONS.
C                               UNITS: CM2
C                               1ST DIMENSION: ENERGY INDEX
C                               2ND DIMENSION: INDEXED BY I4IDFL(N,L)
C  INPUT : (R*8)  SIGMA(,)  = READ - M-RESOLVED CHARGE EXCHANGE
C                                     CROSS-SECTIONS.
C                               UNITS: CM2
C                               1ST DIMENSION: ENERGY INDEX
C                               2ND DIMENSION: INDEXED BY I4IDFM(N,L,M)
C                                              WITH M >= 0 ONLY
C          (I*4)  NWIDTH   = NUMBER OF ENERGY VALUES PER LINE
C          (I*4)  IVALUE   = USED TO PARSE FOR END OF DATA FLAG (-1).
C          (I*4)  N        = N QUANTUM NUMBER.
C          (I*4)  L        = L QUANTUM NUMBER.
C          (I*4)  M        = M QUANTUM NUMBER.
C          (I*4)  I        = LOOP COUNTER.
C          (I*4)  K        = LOOP COUNTER.
C          (I*4)  IERR     = ERROR RETURN CODE.
C          (C*1)  INDD     = DONOR STATE INDEX.
C          (C*28) UID      = USER IDENTIFIER.
C          (C*8)  DATE     = CURRENT DATE.
C          (R*8)  FMUL     = MULTIPLER 1.384D4 * 100
C
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
```

```
C          ------------------------------------------------------------
C          I4FCTN     ADAS      RETURNS CHARACTER STRING AS AN INTEGER.
C          I4UNIT     ADAS      FETCH UNIT NUMBER FOR OUTPUT OF MESSAGES
C          I4IDFL     ADAS      RETURNS UNIQUE INDEX FROM QUANTUM
C                               NUMBERS N AND L.
C          I4IDFM     ADAS      RETURNS UNIQUE INDEX FROM QUANTUM
C                               NUMBERS N, L AND M.
C          XXIDTL     ADAS      INVERSE OF I4IDFL. RETURNS QUANTUM
C                               NUMBERS N AND L FROM INDEX.
C          XXIDTM     ADAS      INVERSE OF I4IDFM. RETURNS QUANTUM
C                               NUMBERS N, L AND M FROM INDEX.
C          XXNAME     ADAS      FINDS REAL NAME OF USER
C          XXSLEN     ADAS      FINDS NON BLANK PART OF STRING
C
C
C AUTHOR:  H. P. SUMMERS, UNIVERSITY OF STRATHCLYDE
C          JA8.08
C          TEL. 0141-553-4196
C DATE:    19/09/95
C
C
C UPDATE:  27/08/97  HP SUMMERS - CHANGED NAME FROM CCWR11 TO CDWR11
C                                 REMOVED REDUNDANT FORMATS 2000, 2006
C UPDATE:  09/07/98  Martin O'Mullane  - added DATE to input list and
C                                        removed call to xxuid
C
C VERSION: 1.1 DATE: 01-12-98
C MODIFIED: RICHARD MARTIN
C - PUT UNDER SCCS CONTROL
C
C VERSION:      1.2                                        DATE: 24-03-99
C MODIFIED: MARTIN O'MULLANE
C               - SECOND VERSION
C
C VERSION:      1.3                                        DATE: 17-05-07
C MODIFIED: Allan Whiteford
C               - Updated comments as part of subroutine documentation
C                 procedure.
C
C VERSION  : 1.4
C DATE     : 22-05-2007
C MODIFIED : Martin O'Mullane
C              - Remove unused m-subshell data possibility.
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      CHARACTER*2         CATYP
      CHARACTER*8         DATE
      CHARACTER*80        DSFULL
      CHARACTER*2         SYMBD,       SYMBR
      INTEGER             INDD,        IUNIT,       IZD,         IZR
      INTEGER             LFORMA(MXNENG),           MXNENG,      MXNSHL
      INTEGER             NENRGY,      NMAX,        NMIN
      LOGICAL             LPARMS,      LSETL
      REAL*8              ALPHAA(MXNENG),           AMDON,       AMREC
      REAL*8              DREN,        ENRGYA(NENRGY)
      REAL*8              PL2A(MXNENG),             PL3A(MXNENG)
      REAL*8              SIGLA(MXNENG,(MXNSHL*(MXNSHL+1))/2)
      REAL*8              SIGNA(MXNENG,MXNSHL),     SIGTA(MXNENG)
      REAL*8              XLCUTA(MXNENG)
```

4.73 cewr12: Subroutine cewr12 from library adas3xx

```
       SUBROUTINE CEWR12( IUNIT  , MXNENG , MXNSHL , ILTYP  ,
     &                    DSFULL , DATE   ,
     &                    CATYP  ,
     &                    SYMBR  , SYMBD  , IZR    , IZD    ,
     &                    INDD   , NENRGY , NMIN   , NMAX   ,
     &                    LPARMS , LSETL  , LSETM  , ENRGYA ,
     &                    ALPHAA , LFORMA , XLCUTA , PL2A   ,
     &                    PL3A   , SIGTA  , SIGNA  , SIGLA
     &                  )
C-----------------------------------------------------------------------
C  ****************** FORTRAN77 SUBROUTINE: CEWR12 *********************
C
C  PURPOSE:  TO OUTPUT DATA TO ADF24 FILE.
C
C  CALLING PROGRAM: ADAS314
C  DATA:
C           THE UNITS USED IN THE DATA FILE ARE TAKEN AS FOLLOWS:
C           COLLISION ENERGIES  : KEV/AMU
C           ALPHA               :
C           TOTAL XSECTS.       : CM2
C           N-SHELL XSECTS.     : CM2
C           NL-SHELL DATA       : CM2
C           NLM-SHELL DATA      : CM2
C  SUBROUTINE:
C  INPUT : (I*4)  IUNIT     = INPUT  UNIT NUMBER FOR RESULTS
C  INPUT : (I*4)  MXNENG    = MAXIMUM NO. OF ENERGIES.
C  INPUT : (I*4)  MXNSHL    = MAXIMUM NO. OF N SHELLS.
C  INPUT : (I*4)  ILTYPL    = TYPE FOR LOW ENERGY X-SECT. EXTRAPOLATION
C  INPUT : (C*80) TITLED    = NOT SET - TITLE FOR DATA SOURCE.
C  INPUT : (C*44) DSFULL    = SOURCE DATASET
C  INPUT : (C2)   CATYP     = 'TT' THERMAL/THERMAL (EQUAL TEMPERATURES
C                                   FOR DONOR AND RECEIVER ONLY)
C                             'TR' THERMAL RECEIVER, MONOENERGETIC DONOR
C                             'TD' THERMAL DONOR, MONOENERGETIC RECEIVER
C                             'ME' SPECIAL MONOENERGETIC CASE
C  INPUT : (C*2)  SYMBR     = READ - RECEIVER ION ELEMENT SYMBOL.
C  INPUT : (C*2)  SYMBD     = READ - DONOR ION ELMENT SYMBOL.
C  INPUT : (I*4)  IZR       = READ - ION CHARGE OF RECEIVER.
C  INPUT : (I*4)  IZD       = READ - ION CHARGE OF DONOR.
C  INPUT : (I*4)  INDD      = READ - DONOR STATE INDEX.
C  INPUT : (I*4)  NENRGY    = NUMBER OF ENERGIES READ.
C  INPUT : (I*4)  NMIN      = LOWEST N-SHELL FOR WHICH DATA READ.
C  INPUT : (I*4)  NMAX      = HIGHEST N-SHELL FOR WHICH DATA READ.
C  INPUT : (L*4)  LPARMS    = FLAGS IF L-SPLITTING PARAMETERS PRESENT.
C                             .TRUE.  => L-SPLITTING PARAMETERS PRESENT.
C                             .FALSE  => L-SPLITTING PARAMETERS ABSENT.
C  INPUT : (L*4)  LSETL     = FLAGS IF L-RESOLVED DATA PRESENT.
C                             .TRUE.  => L-RESOLVED DATA PRESENT.
C                             .FALSE  => L-RESOLVED DATA ABSENT.
C  INPUT : (L*4)  LSETM     = FLAGS IF M-RESOLVED DATA PRESENT.
C                             .TRUE.  => M-RESOLVED DATA PRESENT.
C                             .FALSE  => M-RESOLVED DATA ABSENT.
C  INPUT : (R*8)  ENRGYA()  = READ - COLLISION ENERGIES.
C                             UNITS: EV/AMU (READ AS KEV/AMU)
C                             DIMENSION: ENERGY INDEX
C  INPUT : (R*8)  ALPHAA()  = READ - EXTRAPOLATION PARAMETER ALPHA.
C                             DIMENSION: ENERGY INDEX
C  INPUT : (I*4)  LFORMA()  = READ - PARAMETERS FOR CALCULATING L-RES
C                                    X-SEC.
```

```
C                                DIMENSION: ENERGY INDEX
C  INPUT : (R*8)  XLCUTA()  = READ - PARAMETERS FOR CALCULATING L-RES
C                                       X-SEC.
C                                DIMENSION: ENERGY INDEX
C  INPUT : (R*8)  PL2A()    = READ - PARAMETERS FOR CALCULATING L-RES
C                                       X-SEC.
C                                DIMENSION: ENERGY INDEX
C  INPUT : (R*8)  PL3A()    = READ - PARAMETERS FOR CALCULATING L-RES
C                                       X-SEC.
C                                DIMENSION: ENERGY INDEX
C  INPUT : (R*8)  SIGTA()   = READ - TOTAL CHARGE EXCHANGE
C                                       CROSS-SECTION.
C                                UNITS: CM2
C                                DIMENSION: ENERGY INDEX
C  INPUT : (R*8)  SIGNA(,)  = READ - N-RESOLVED CHARGE EXCHANGE
C                                       CROSS-SECTIONS.
C                                UNITS: CM2
C                                1ST DIMENSION: ENERGY INDEX
C                                2ND DIMENSION: N-SHELL
C  INPUT : (R*8)  SIGLA(,)  = READ - L-RESOLVED CHARGE EXCHANGE
C                                       CROSS-SECTIONS.
C                                UNITS: CM2
C                                1ST DIMENSION: ENERGY INDEX
C                                2ND DIMENSION: INDEXED BY I4IDFL(N,L)
C  INPUT : (R*8)  SIGMA(,)  = READ - M-RESOLVED CHARGE EXCHANGE
C                                       CROSS-SECTIONS.
C                                UNITS: CM2
C                                1ST DIMENSION: ENERGY INDEX
C                                2ND DIMENSION: INDEXED BY I4IDFM(N,L,M)
C                                               WITH M >= 0 ONLY
C          (I*4)  NWIDTH   = NUMBER OF ENERGY VALUES PER LINE
C          (I*4)  IBLK     = CURRENT DATA BLOCK.
C          (I*4)  IVALUE   = USED TO PARSE FOR END OF DATA FLAG (-1).
C          (I*4)  N        = N QUANTUM NUMBER.
C          (I*4)  L        = L QUANTUM NUMBER.
C          (I*4)  M        = M QUANTUM NUMBER.
C          (I*4)  I        = LOOP COUNTER.
C          (I*4)  K        = LOOP COUNTER.
C          (I*4)  IERR     = ERROR RETURN CODE.
C          (R*8)  ALPH0    = LOW ENERGY PARAMETER FOR ILTYP = 1
C          (C*1)  INDD     = DONOR STATE INDEX.
C          (C*9)  FST      = FINAL STATE NAME.
C          (C*9)  BLK9     = BLANK STRING OF LENGTH 9.
C          (C*1)  LCHRA()  = CHARACTER FOR L ANG.MOM.INDEXED BY L+1
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4FCTN     ADAS      RETURNS CHARACTER STRING AS AN INTEGER.
C          I4UNIT     ADAS      FETCH UNIT NUMBER FOR OUTPUT OF MESSAGES
C          I4IDFL     ADAS      RETURNS UNIQUE INDEX FROM QUANTUM
C                               NUMBERS N AND L.
C          I4IDFM     ADAS      RETURNS UNIQUE INDEX FROM QUANTUM
C                               NUMBERS N, L AND M.
C          XXIDTL     ADAS      INVERSE OF I4IDFL. RETURNS QUANTUM
C                               NUMBERS N AND L FROM INDEX.
C          XXIDTM     ADAS      INVERSE OF I4IDFM. RETURNS QUANTUM
C                               NUMBERS N, L AND M FROM INDEX.
C          XXNAME     ADAS      FINDS REAL NAME OF USER
C          XXSLEN     ADAS      FINDS NON BLANK PART OF STRING
C AUTHOR:  H. P. SUMMERS, UNIVERSITY OF STRATHCLYDE
C          JA8.08
```

```
C          TEL. 0141-553-4196
C DATE:    13/11/95
C UPDATE:  27/08/97  HP SUMMERS - CHANGED NAME FROM CCWR12 TO CDWR12
C UPDATE:  09/07/98  Martin O'Mullane  - added DATE to input list and
C                                       removed call to xxuid
C
C VERSION: 1.1 DATE: 01-12-98
C MODIFIED: RICHARD MARTIN
C - PUT UNDER SCCS CONTROL
C
C VERSION: 1.2 DATE: 17-05-07
C MODIFIED: Allan Whiteford
C - Updated comments as part of subroutine documentation
C                 procedure.
C
C VERSION  : 1.3
C DATE     : 22-05-2007
C MODIFIED : Martin O'Mullane
C              - Remove unused m-subshell data possibility.
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      CHARACTER*2         CATYP
      CHARACTER*8         DATE
      CHARACTER*80        DSFULL
      CHARACTER*2         SYMBD,        SYMBR
      INTEGER             ILTYP,        INDD,         IUNIT,        IZD
      INTEGER             IZR,          LFORMA(MXNENG),             MXNENG
      INTEGER             MXNSHL,       NENRGY,       NMAX,         NMIN
      LOGICAL             LPARMS,       LSETL,        LSETM
      REAL*8              ALPHAA(MXNENG),             ENRGYA(NENRGY)
      REAL*8              PL2A(MXNENG),               PL3A(MXNENG)
      REAL*8              SIGLA(MXNENG,(MXNSHL*(MXNSHL+1))/2)
      REAL*8              SIGNA(MXNENG,MXNSHL),       SIGTA(MXNENG)
      REAL*8              XLCUTA(MXNENG)
```

4.74 cldlbn2: Subroutine cldlbn2 from library adas3xx

```
       SUBROUTINE CLDLBN2(exfile,Z0,Z1,ZEFF,DENS,TE,DENSP,TP,BMENER,
     &                    DENSH,W1,NMIN,NMAX,NREP,IMAX,ARED,
     &                    RHS,CIONPT,DRECPT,RRECPT,XRECPT,IECION,
     &                    IEDREC,IERREC,IEXREC,DVEC,ACNST,A1CNST,
     &                    lpass)
C
       IMPLICIT REAL*8(A-H,O-Z)
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: CLDLBN2 *******************
C
C-----------------------------------------------------------------------
C  PURPOSE: ESTABLISH THE PROJECTED INFLUENCE OF HIGH N-SHELLS IN
C  THE BUNDLE-N COLLISIONAL DIELECTRONIC MODEL ON A SET OF LS OR LSJ
C  RESOLVED LOW LEVEL POPULATION EQUATIONS
C
C  BOTH THE RECOMBINATION AND IONISATION PATHWAYS THROUGH THE HIGH
C  LEVELS ARE TAKEN INTO ACCOUNT AS WELL AS THE INDIRECT COUPLINGS OF
C  LOW RESOLVED LEVELS VIA THE HIGH BUNDLE-N LEVELS.
C
C  THE SUBROUTINE IS USED AS AN ARBITRARY CALL FROM WITHIN THE
C  CONVENTIONAL BNDLEN ROUTINE FOLLOWING ESTABLISHMENT OF THE
C  CONDENSED COLLISIONAL-DIELECTRONIC MATRIX AND RIGHT-HAND SIDE
C
C  THE EXPANSION OF THE PROJECTED MATRICES OVER THE RESOLVED LOWER
C  LEVELS IS DEFINED THROUGH RESOLVED - BUNDLEN INDEXING AND
C  WEIGHTING FRACTION TABLES STORED IN DATA STATEMENTS.
C  THESE ARE HELD FOR COMBINATIONS BASED ON THE A-D-A-S DATA BASE
C  MEMBERS.
C
C  THE ROUTINE PROVIDES TABULAR OUTPUT AND FOR THE MOMENT PREPARES A
C  PASSING FILE FOR FURTHER PROCESSING IN THE A-D-A-S STRUCTURE
C
C  PROCESSING CONTINUES WITH EXECUTION OF THE LOW LEVEL POPULATION
C  CALCULATION PROVIDED THE LOW LEVEL DATA FILE 'REFMEM' IS NON-BLANK
C  OTHERWISE ONLY THE PASSING FILE IS PRODUCED
C
C  INPUT
C      exfile   = DATA SET NAME OF EXPANSION FILE
C      Z0       = NUCLEAR CHARGE
C      Z1       = RECOMBINING ION CHARGE
C      ZEFF     = PLASMA Z EFFECTIVE
C      DENS     = ELECTRON DENSITY (CM-3)
C      TE       = ELECTRON TEMPERATURE (K)
C      DENSP    = PROTON   DENSITY (CM-3)
C      TP       = PROTON   TEMPERATURE (K)
C      BMENER   = NEUTRAL HYDROGEN BEAM ENERGY (EV/AMU)
C      DENSH    = NEUTRAL BEAM HYDROGEN DENSITY (CM-3)
C      W1       = GROUND STATE RADIATION DILUTION FACTOR
C      NMIN     = LOWEST N-SHELL
C      NMAX     = HIGHEST N-SHELL
C      NREP(I)  = SET OF REPRESENTATIVE LEVELS
C      IMAX     = NUMBER OF REPRESENTATIVE LEVELS
C      ARED(I,J)= CONDENSED COLLISONAL-DIELECTRONIC MATRIX (CN SOLUTION)
C      RHS(I)   = CONDENSED RIGHT-HAND-SIDE                (CN SOLUTION)
C      CIONPT(I)= COLLISIONAL IONISATION CONTRIBUTION TO ARED(I,I)
C      DRECPT(I)= DIELECTRONIC RECOMBINATION CONTRIBUTION TO RHS(I)
C      RRECPT(I)= RADIATIVE    RECOMBINATION CONTRIBUTION TO RHS(I)
```

```
C      XRECPT(I)= CHARGE EXCHANGE RECOMB.    CONTRIBUTION TO RHS(I)
C      IECION   = 0 ELIMINATE CIONPT FOR LOW LEVELS IN PROJECTION
C                 1 DO NOT ELIMINATE CIONPT.
C      IEDREC   = 0 ELIMINATE DRECPT FOR LOW LEVEL PROJECTION
C                 1 DO NOT ELIMINATE DRECPT.
C      IERREC   = 0 ELIMINATE RRECPT FOR LOW LEVEL PROJECTION
C                 1 DO NOT ELIMINATE RRECPT.
C      IEXREC   = 0 ELIMINATE XRECPT FOR LOW LEVEL PROJECTION
C                 1 DO NOT ELIMINATE XRECPT.
C      DVEC(I)  = CONVERSION FACTOR FOR BN --> POPULATION
C      ACNST    = 1.03928D-13*Z*ATE*DSQRT(ATE)
C      A1CNST   = 6.60074D-24*DENS*(157890.0/TE)**1.5
C
C  OUTPUT
C
C
C  ********** H.P. SUMMERS, JET               8 FEB 1990    ************
C  **********                                24 APR 1990    ************
C  **********                                18 JUL 1991    ************
C----------------------------------------------------------------------
C
C----------------------------------------------------------------------
C
C UPDATE:  19/01/94 - JONATHAN NASH - TESSELLA SUPPORT SERVICES PLC
C
C          THE FOLLOWING MODIFICATIONS HAVE BEEN MADE TO THE SUBROUTINE:
C
C          1) THE COMPLETE EXPANSION FILE DATA SET NAME IS NOW PASSED
C             INTO THE ROUTINE RATHER THAN JUST THE MEMBER NAME.
C
C          2) THE OUTPUT FILE (UNIT 18) IS NOW OPENED EXTERNAL TO THIS
C             ROUTINE.
C
C  NOTES:  NO ATTEMPT HAS BEEN MADE TO RESTRUCTURE THE ROUTINE. RATHER
C          THE MINIMUM AMOUNT OF WORK TO INTEGRATE THE ROUTINE INTO
C          ADAS310 HAS BEEN COMPLETED.
C
C----------------------------------------------------------------------
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 16-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST VERSION
C
C VERSION: 1.2                          DATE: 17-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - CHANGED ACTION=READ TO STATUS=UNKNOWN IN OPEN
C                 STATEMENT
C
C VERSION: 1.3                          DATE: 24-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - COMMENTED OUT CALL TO USINFO AS NO LONGER NEEDED
C
C VERSION: 1.4                          DATE: 08-02-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - REMOVED SUPERFLUOUS VARIABLES
C
C VERSION: 1.5                          DATE: 16-05-07
C MODIFIED: Allan Whiteford
C               -  Updated comments as part of subroutine documentation
```

```
C                  procedure.
C
C VERSION : 1.6
C DATE    : 26-06-2007
C MODIFIED: Hugh Summers
C               - Revised for heavy species cx. Changed
C                 exmemb to full file name exfile.
C               - Add LPASS to turn on/off output to unit 18 depending on
C                 calling routine; adas310 and adas316 have different
C                 approaches to file management.
C
C-----------------------------------------------------------------------
C
      CHARACTER*(*)       EXFILE
      INTEGER             IECION,      IEDREC,      IERREC,      IEXREC
      INTEGER             IMAX,        NMAX,        NMIN
      INTEGER             NREP(NDIM+1)
      LOGICAL             LPASS
      REAL*8              A1CNST,      ACNST,       ARED(NDIM,NDIM)
      REAL*8              BMENER,      CIONPT(NDIM),             DENS
      REAL*8              DENSH,       DENSP,       DRECPT(NDIM)
      REAL*8              DVEC(NDIM),  RHS(NDIM),   RRECPT(NDIM)
      REAL*8              TE,          TP,          W1
      REAL*8              XRECPT(NDIM),             Z0,          Z1
      REAL*8              ZEFF
```

4.75 cmprss: Subroutine cmprss from library adas3xx

```
      SUBROUTINE CMPRSS(IUIN,IUOUT,DATE,IECOUNT,INCOUNT,ITCOUNT)
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: CMPRSS ********************
C
C PURPOSE:
C  READ OUTPUT FROM V2BNLDN1 AND PRODUCE TABLES OF BEAM STOPPING RATES
C  AS A FUNCTION OF PLASMA DENSITY AND TEMPERATURE AND OF BEAM ENERGY
C  (FOR USE IN KS4FIT BY QHIOCH)
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
C
C UPDATE:  20/01/94 - JONATHAN NASH - TESSELLA SUPPORT SERVICES PLC
C
C          THIS SUBROUTINE WAS ADAPTED FROM THE PROGRAM
C          'JETXJS.BMSTOP.FORT(COMPRESS)'.
C
C          THE FOLLOWING MODIFICATIONS HAVE BEEN MADE:
C
C          1) THE MAXIMUM NUMBER OF BEAM ENERGIES HAS BEEN INCREASED
C             FROM 13 TO 20.
C
C          2) THE UNIT NUMBERS OF THE INPUT AND OUTPUT FILES ARE PASSED
C             INTO THE ROUTINE AS ARGUMENTS.
C
C          3) THE REFERENCE ELECTRON DENSITY AND TEMPERATURE AND BEAM
C             ENERGY ARE DETERMINED RATHER THAN BEING DECLARED AS
C             PARAMETERS. THESE ASSIGMENTS ARE BASED ON KNOWING THE
C             ORDER OF THE DATA IN THE INPUT FILE.
C
C  NOTES:  NO ATTEMPT HAS BEEN MADE TO RESTRUCTURE THE ROUTINE. RATHER
C          THE MINIMUM AMOUNT OF WORK TO INTEGRATE THE ROUTINE INTO
C          ADAS310 HAS BEEN COMPLETED.
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 01-02-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - PUT UNDER SCCS CONTROL
C
C VERSION: 1.2                          DATE: 01-02-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - ALLOWED FOR CASE WHERE ZEFF=0.0
C
C VERSION: 1.3                          DATE: 05-02-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - ADDED NUMBER OF ENERGIES, TEMPS. AND DENSITIES
C                 AS OUTPUT.
C
C VERSION: 1.4                          DATE: 05-02-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - CORRECTED SCCS KEYWORDS.
C
C VERSION: 1.5                          DATE: 08-02-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - REMOVED SUPERFLUOUS VARIABLES
```

```
C
C VERSION: 1.6                          DATE: 14-02-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - CHANGED FORMAT 1004 TO READ 97X RATHER THAN
C                 98X AS THIS WAS NOT BEING READ CORRECTLY
C
C VERSION: 1.7                          DATE: 23-09-96
C MODIFIED: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C               - CHANGED FORMAT 1004 TO READ E12.5 RATHER THAN
C                 E11.5 AS THIS WAS NOT BEING READ CORRECTLY
C VERSION: 1.8
C MODIFIED: HARVEY ANDERSON
C DATE: 23/01/96
C - MODIFIED THE VALUE OF MAXTE, MAXNE, AND MAXEB FROM 10
C   TO 25.
C
C VERSION: 1.9
C MODIFIED: HARVEY ANDERSON
C DATE: 09/08/99
C - EXTENDED THE ARRAY CONTAINING THE CHEMICAL ELEMENT
C   SYMBOL TO INCLUDE ALL SPECIES UP TO ZN.
C
C
C-----------------------------------------------------------------------
C
C  INPUT:  (I*4)  IUIN      = UNIT NO. OF INPUT FILE.
C  INPUT:  (I*4)  IUOUT     = UNIT NO. OF OUTPUT FILE.
C  INPUT:  (C*8)  DATE      = DATE STRING.
C  OUTPUT: (I*4)  IECOUNT   = NUMBER OF BEAM ENERGIES
C  OUTPUT: (I*4)  INCOUNT   = NUMBER OF DENSITIES
C  OUTPUT: (I*4)  ITCOUNT   = NUMBER OF TEMPERATURES
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      CHARACTER*8         DATE
      INTEGER             IECOUNT,     INCOUNT,     ITCOUNT,     IUIN
      INTEGER             IUOUT
```

4.76 colint: Subroutine colint from library adas3xx

```
       SUBROUTINE COLINT(Y,Z,EN,AI)
C
       IMPLICIT REAL*8 (A-H,O-Z)
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: COLINT *********************
C
C PURPOSE UNKNOWN
C
C NOTES: THIS ROUTINE IS NOT YET PROPERLY ANNOTATED
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 16-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST VERSION
C
C VERSION: 1.2                          DATE: 08-02-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - REPLACED 0.0 ARGUMENT IN CALL TO YIP
C                 WITH A DUMMY VARIABLE INSTEAD.
C
C-----------------------------------------------------------------------
C
      REAL*8              AI,          EN,          Y,           Z
```

4.77 cxbms: Subroutine cxbms from library adas3xx

```
      SUBROUTINE CXBMS(DSNIN,NSITYP,IOUNIT,SIFRAC,UBMENG,UTDENS,
     &           UTTEMP,NREQ,MXREQS,BSTOT)

C-----------------------------------------------------------------------
C
C       ********** FORTRAN 77 SUBROUTINE: CXBMS **********
C
C PURPOSE: TO ASSEMBLE COMPOSITE BEAM STOPPING OF EMISSION
C  COEFFICIENTS USING THE LINEAR INTERPOLATION
C  AND COMBINATION METHOD.
C
C
C INPUT
C
C (R*8)  UBMENG : USER REQUESTED NEUTRAL BEAM ENERGIES
C   UNITS: EV/AMU
C   DIMENSION: NREQ
C (R*8)  UTDENS : USER REQUESTED TARGET DENSITIES
C   UNITS: CM-3
C   DIMENSION: NREQ
C (R*8)  UTTEMP : USER REQUESTED TARGET TEMPERATURES
C    UNITS: EV
C   DIMENSION: NREQ
C (R*8)  SIFRAC   : FRACTIONAL IMPURITY CONTENT.
C   1ST DIMENSION: MXREQS
C                          2ND DIMENSION: NSITYP
C (I*4)  NSITYP : NUMBER OF PLASMA IMPURITY IONS.
C (I*4)  NREQ : NUMBER OF REQUESTED ENERGIES,
C           DENSITIES AND TEMPERATURES.
C (I*4)  MXREQS   : SIZE OF FIRST DIMENSION OF SIFRAC
C
C (I*4)  IOUNIT : UNIT NUMBER EMPLOYED TO READ ADF21
C   AND ADF22 TYPE FILES.
C
C (CHR)  DSNIN() : ARRAY CONTAINING NAME OF EACH FILE TO BE READ.
C                          DIMENSION: NSITYP
C
C
C       OUTPUT
C
C       (R*8)  BSTOT()  : TOTAL BEAM STOPPING COEFFICIENTS.
C                         DIMENSION: NREQ
C
C GENERAL
C
C       (R*8)  BEREF()  : REFERENCE BEAM ENERGIES.
C                         UNITS: EV/AMU
C                         DIMENSION: MXIT
C       (R*8)  TDREF()  : REFERENCE TARGET DENSITIES.
C                         UNITS: CM-3
C                         DIMENSION: MXIT
C       (R*8)  TTREF()  : REFERENCE TARGET TEMPERATURES.
C                         UNITS: EV
C                         DIMENSION: MXIT
C       (R*8)  SVREF()  : STOPPING COEFFT. AT REFERENCE BEAM ENERGY,
C                         TARGET DENSITY AND TEMPERATURE.
C                         UNITS: CM3 S-1
C                         DIMENSION: MXIT
C       (R*8)  BE(,)    : BEAM ENERGIES.
```

```
C                         UNITS: EV/AMU
C                         1ST DIMENSION: MXBE
C                         2ND DIMENSION: MXIT
C       (R*8)  TDENS(,) : TARGET DENSITIES.
C                         UNITS: CM-3
C                         1ST DIMENSION: MXTD
C                         2ND DIMENSION: MXIT
C       (R*8)  TTEMP(,) : TARGET TEMPERATURES.
C                         UNITS: EV
C                         1ST DIMENSION: MXTT
C                         2ND DIMENSION: MXIT
C       (R*8)  SVT(,)   : STOPPING COEFFT. AT REFERENCE BEAM ENERGY
C                         AND TARGET DENSITY.
C                         UNITS: CM3 S-1
C                         1ST DIMENSION: MXTT
C                         2ND DIMENSION: MXIT
C       (R*8)  SVED(,,) : STOPPING COEFFT. AT REFERENCE TARGET
C                         TEMPERATURE.
C                         UNITS: CM3 S-1
C                         1ST DIMENSION: MXBE
C                         2ND DIMENSION: MXTD
C                         3RD DIMENSION: MXIT
C       (R*8)  BSION(,) : BEAM STOPPING COEFFICIENTS FOR INDIVIDUAL
C                         IONS.
C                         1ST DIMENSION: MXREQ
C                         2ND DIMENSION: MXIT
C
C (R*8)  FACT1    : FACTOR USED IN CALCULATING ZEFF
C (R*8)  FACT2    : SIMILAR TO FACT1.
C (R*8)  ZEFF() : USED SO THAT THE BEAM STOPPING
C                         COEFFICIENTS FOR INDIVUDUAL
C   IMPURITY IONS ARE EVALUATED AT THE
C                         CORRECT EFFECTIVE ELECTRON DENSITY.
C (R*8)  WT : WEIGHTING FACTOR.
C (R*8)  TFRAC : GENERAL VARIABLE.
C
C (I*4)  MXREQ    : MAXIMUM NUMBER OF REQUESTED ENERGIES
C   DENSITIES AND TEMPERATURES
C (I*4)  MXIT     : MAXIMUM POSSIBLE NUMBER OF DIFFERENT
C   PLASMA IONS.
C (I*4)  MXBE : MAXIMUM NUMBER OF BEAM ENERGIES WHICH CAN
C   BE READ FROM THE ADF21/22 TYPE FILES.
C (I*4)  MXTD : MAXIMUM NUMBER OF TARGET DENSITIES WHICH CAN
C   BE READ FROM THE ADF21/22 TYPE FILES.
C (I*4)  MXTT : MAXIMUM NUMBER OF TARGET TEMPERATURES WHICH
C   CAN BE READ FROM THE ADF21/22 TYPE FILES.
C       (I*4)  ITZ()    : TARGET ION CHARGE.
C                         DIMENSION: MXIT
C       (I*4)  NBE()    : NUMBER OF BEAM ENERGIES.
C                         DIMENSION: MXIT
C       (I*4)  NTDENS() : NUMBER OF TARGET DENSITIES.
C                         DIMENSION: MXIT
C       (I*4)  NTTEMP() : NUMBER OF TARGET TEMPERATURES.
C                         DIMENSION: MXIT
C
C       (CHR)  TSYM()   : TARGET ION ELEMENT SYMBOL.
C                         DIMENSION: MXIT
C
C       (L*4)  LIBMA(,) : FLAGS IF INTERPOLATION OR EXTRAPOLATION
C                         USED FOR REQUESTED BEAM ENERGIES.
C                         .TRUE.  => INTERPOLATION USED.
```

```
C                          .FALSE. => EXTRAPOLATION USED.
C                          1ST DIMENSION: MXREQ
C                          2ND DIMENSION: MXIT
C   NOTE: USE OF FLAGS NOT IMPLEMENTED
C       (L*4)  LIDNA(,) : FLAGS IF INTERPOLATION OR EXTRAPOLATION
C                          USED FOR REQUESTED ION DENSITIES.
C                          .TRUE.  => INTERPOLATION USED.
C                          .FALSE. => EXTRAPOLATION USED.
C                          1ST DIMENSION: MXREQ
C                          2ND DIMENSION: MXIT
C   NOTE: USE OF FLAGS NOT IMPLEMENTED
C       (L*4)  LITIA(,) : FLAGS IF INTERPOLATION OR EXTRAPOLATION
C                          USED FOR REQUESTED ION TEMPERATURES.
C                          .TRUE.  => INTERPOLATION USED.
C                          .FALSE. => EXTRAPOLATION USED.
C                          1ST DIMENSION: MXREQ
C                          2ND DIMENSION: MXIT
C   NOTE: USE OF FLAGS NOT IMPLEMENTED
C
C  ROUTINES:
C
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          C4DATA     ADAS      READS INPUT DATA SET IN ADF21/22 FORMAT.
C          C4SPLN     ADAS      PERFORMS SPLINE ON INPUT DATA.
C          I4UNIT     ADAS      RETURNS UNIT NO. FOR OUTPUT OF MESSAGES.
C
C
C  NOTE  : THE IMPURITY FRACTIONS ARE RENORMNALISED TO
C ENSURE THAT THE TOTAL SUM OF EACH IMPURITY
C         FRACTION DOES NOT EXCEED THE VALUE OF ONE.
C
C
C   AUTHOR: HARVEY ANDERSON
C         UNIVERSITY OF STRATHCLYDE
C ANDERSON@PHYS.STRATH.AC.UK
C
C DATE  : 30/09/99
C
C VERSION : 1.1
C       DATE    : 3-6-2000
C MODIFIED: Martin O'Mullane
C   Repositioned declaration of variables into standard
C                ADAS convention and to satisfy g77.
C
C VERSION : 1.2
C       DATE    : 19-3-2003
C MODIFIED: Lorne Horton
C   Increased MXREQ to 1024.  Added check on internal
C                matrix sizes. Added LSET flag to allow faster
C                splines on repeated calls to C4SPLN.
C
C VERSION : 1.3
C       DATE    : 03-12-2004
C MODIFIED: Martin O'Mullane
C    - Replace c4data with xxdata_21.
C    - Increase dsnin length to 132 characters.
C    - Merge L Horton's changes into central version.
C
C VERSION : 1.4
C       DATE    : 08-12-2004
```

```
C MODIFIED: Martin O'Mullane/Allan Whiteford
C    - Increase size of dsnsav to 132 and set initial
C                    values to ' ' rather than ''.
C
C-----------------------------------------------------------------------
      CHARACTER*132       DSNIN(NSITYP)
      INTEGER             IOUNIT,      MXREQS,      NREQ,        NSITYP
      REAL*8              BSTOT(NREQ), SIFRAC(MXREQS,NSITYP)
      REAL*8              UBMENG(NREQ),             UTDENS(NREQ)
      REAL*8              UTTEMP(NREQ)
```

4.78 cxchrg: Subroutine cxchrg from library adas3xx

```
      SUBROUTINE CXCHRG( SYMBD , IZD   , SYMBR , IZR  , IDZ0  ,
     &                   IRZ0  , IRZ1  , IRZ2
     &                 )
C
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: CXCHRG *********************
C
C  PURPOSE:  SETS UP NUCLEAR CHARGE OF DONOR AND NULEAR, INITIAL AND
C            FINAL CHARGES OF RECEIVER.
C
C  CALLING PROGRAM: C6CHRG , ADAS308
C
C  INPUT : (C*2)  SYMBD   = DONOR ELEMENT SYMBOL.
C  INPUT : (I*4)  IZD     = DONOR ION CHARGE.
C  INPUT : (C*2)  SYMBR   = RECEIVER ELEMENT SYMBOL.
C  INPUT : (I*4)  IZR     = RECEIVER ION CHARGE.
C
C  OUTPUT: (I*4)  IDZ0    = DONOR NUCLEAR CHARGE.
C  OUTPUT: (I*4)  IRZ0    = RECEIVER NUCLEAR CHARGE.
C  OUTPUT: (I*4)  IRZ1    = RECEIVER ION INITIAL CHARGE.
C  OUTPUT: (I*4)  IRZ2    = RECEIVER ION FINAL CHARGE.
C
C
C ROUTINES:
C           ROUTINE     SOURCE      BRIEF DESCRIPTION
C           ------------------------------------------------------------
C           I4IEZ0      ADAS        RETURNS THE NUCLEAR CHARGE GIVEN THE
C                                   ELEMENT SYMBOL.
C           XXSTUC      ADAS        ENSURES ALL LETTERS IN STRING ARE
C                                   UPPER CASE.
C
C AUTHOR:  JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 5183
C
C DATE:    11/11/93
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      CHARACTER*2         SYMBD,       SYMBR
      INTEGER             IDZ0,        IRZ0,        IRZ1,        IRZ2
      INTEGER             IZD,         IZR
```

4.79 cxcrdg: Subroutine cxcrdg from library adas3xx

```
      SUBROUTINE CXCRDG ( MXNSHL , MXJSHL , IZ0    , IZ1    ,
     &                    NI     , LI     ,  NJ    , LJ     ,
     &                    TEV    , DENS   , ZP     , TPV    ,
     &                    EMP    , TBLF   , GAE    , GAP    ,
     &                    QEP    , QEM    , QIP    , QIM
     &                  )
C
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: CXCRDG *********************
C
C  PURPOSE:  CALCULATES ELECTRON AND +VE ION COLLISIONAL RATE
C            COEFFICIENTS BETWEEN NEARLY DEGENERATE LEVELS FOR
C            HYDROGEN-LIKE, LITHIUM-LIKE AND SODIUM-LIKE IONS.
C
C            RATES ARE CALCULATED FOR BOTH TERM AVERAGED AND
C            SEPARATE NLJ->NL+1J' AND NLJ->NL-1J' STATES.
C
C
C  CALLING PROGRAM: C6TBQM , C8TBQM
C
C  INPUT : (I*4)  MXNSHL  = MAXIMUM NUMBER OF N SHELLS.
C  INPUT : (I*4)  MXJSHL  = MAXIMUM NUMBER OF J SUB-SHELLS.
C  INPUT : (I*4)  IZ0     = TARGET NUCLEAR CHARGE.
C  INPUT : (I*4)  IZ1     = ION CHARGE.
C  INPUT : (I*4)  NI      = VALENCE ELECTRON PRINCIPAL QUANTUM NUMBER
C                           IN STATE I.
C  INPUT : (I*4)  LI      = VALENCE ELECTRON ORBITAL QUANTUM NUMBER IN
C                           STATE I.
C  INPUT : (I*4)  NJ      = VALENCE ELECTRON PRINCIPAL QUANTUM NUMBER
C                           IN STATE J.
C  INPUT : (I*4)  LJ      = VALENCE ELECTRON ORBITAL QUANTUM NUMBER IN
C                           STATE J.
C  INPUT : (R*8)  TEV     = TEMPERATURE (ELECTRON DISTRIBUTION).
C                           UNITS: EV
C  INPUT : (R*8)  DENS    = ELECTRON DENSITY.
C                           UNITS: CM-3
C  INPUT : (R*8)  ZP      = CHARGE OF COLLIDING POSITIVE ION.
C  INPUT : (R*8)  TPV     = TEMPERATURE (COLLIDING POSITIVE ION
C                           DISTRIBUTION).
C                           UNITS: EV
C  INPUT : (R*8)  EMP     = REDUCED MASS FOR COLLIDING POSITIVE ION.
C                           UNITS: ELECTRON MASSES
C  INPUT : (R*8)  TBLF()  = TABLE OF RADIATIVE LIFETIMES.
C                           UNITS: SECS
C                           DIMENSION: REFERENCED BY FUNC I4IDFL(N,L).
C
C  OUTPUT: (R*8)  GAE     = TERM AVERAGED GAMA RATE PARAMETER FOR
C                           ELECTRON COLLISIONS.
C  OUTPUT: (R*8)  GAP     = TERM AVERAGED GAMA RATE PARAMETER FOR
C                           POSITIVE ION COLLISIONS.
C  OUTPUT: (R*8)  QEP()   = ELECTRON RATE COEFFT. FOR NLJ->NL+1J'
C                           DIMENSION: J->J' TRANSITION INDEX.
C  OUTPUT: (R*8)  QEM()   = ELECTRON RATE COEFFT. FOR NLJ->NL-1J'
C                           DIMENSION: J->J' TRANSITION INDEX.
C  OUTPUT: (R*8)  QIP()   = POSITIVE ION RATE COEFFT. FOR NLJ->NL+1J'
C                           DIMENSION: J->J' TRANSITION INDEX.
C  OUTPUT: (R*8)  QIM()   = POSITIVE ION RATE COEFFT. FOR NLJ->NL-1J'
```

```
C                           DIMENSION: J->J' TRANSITION INDEX.
C
C  PARAM : (I*4)  MXTEMP  = MAXIMUM NUMBER OF TEMPERATURES.
C  PARAM : (I*4)  MXCOLL  = MAXIMUM NUMBER OF COLLISION STRENGTHS.
C  PARAM : (I*4)  MXTERM  = 2.
C  PARAM : (R*8)  P1      =
C
C          (I*4)  IZ      = IZ1-1.
C          (I*4)  NTEMP   = NUMBER OF TABULATED TEMPERATURES.
C          (I*4)  NCOLL   = NUMBER OF TABULATED COLLISION STRENGTHS.
C          (I*4)  ICI     =
C          (I*4)  ICJ     =
C          (I*4)  IZC     = INTEGER BELOW CHARGE OF POSITIVE ION.
C          (I*4)  I       = LOOP INDEX.
C          (I*4)  J       = LOOP INDEX.
C
C          (R*8)  WI      = STATISTICAL WEIGHT OF STATE I.
C          (R*8)  WJ      = STATISTICAL WEIGHT OF STATE J.
C          (R*8)  EI      = BINDING ENERGY OF STATE I.
C                           UNITS: RYD
C          (R*8)  EJ      = BINDING ENERGY OF STATE I.
C                           UNITS: RYD
C          (R*8)  TE      = TEMPERATURE (ELECTRON DISTRIBUTION).
C                           UNITS:
C          (R*8)  TP      = TEMPERATURE (COLLIDING POSITIVE ION
C                           DISTRIBUTION).
C                           UNITS:
C          (R*8)  Z1      = REAL VALUE = IZ1.
C          (R*8)  XNI     = REAL VALUE = NI.
C          (R*8)  XLI     = REAL VALUE = LI.
C          (R*8)  XLJ     = REAL VALUE = LJ.
C          (R*8)  XLG     =
C          (R*8)  EI0     = BINDING ENERGY FOR STATE I.
C          (R*8)  EJ0     = BINDING ENERGY FOR STATE J.
C          (R*8)  ZEFFI   = EFFECTIVE ION CHARGE FOR STATE I.
C          (R*8)  ZEFFJ   = EFFECTIVE ION CHARGE FOR STATE J.
C          (R*8)  TAU     =
C          (R*8)  W       =
C          (R*8)  T1      =
C          (R*8)  T2      =
C          (R*8)  XXLI    =
C          (R*8)  XXLJ    =
C          (R*8)  XS      =
C          (R*8)  XXJI    =
C          (R*8)  XXJJ    =
C          (R*8)  DE      =
C          (R*8)  PHI     =
C          (R*8)  ZC      =
C          (R*8)  EMM     =
C          (R*8)  QI1     = COLLISIONAL EXCITATION RATE COEFFICIENT FROM
C                           PENGELLY AND SEATON.
C                           UNITS: CM3 SEC-1
C          (R*8)  QIJ     = COLLISIONAL DEEXCITATION RATE COEFFICIENT
C                           FROM PENGELLY AND SEATON.
C                           UNITS: CM3 SEC-1
C          (R*8)  G1      = GAMMA RATE PARAMETER FROM PENGELLY AND
C                           SEATON.
C                           UNITS:
C          (R*8)  RAT     =
C          (R*8)  SCF     = SCALING FACTOR.
C
```

```
C          (R*8)  EPS()   = INCIDENT ELECTRON ENERGIES.
C                           UNITS: RYD
C                           DIMENSION: COLLISION INDEX.
C          (R*8)  OMEG()  = COLLISION STRENGTHS.
C                           DIMENSION: COLLISION INDEX.
C          (R*8)  TA()    = TEMPERATURES (INCIDENT POSITIVE ION
C                           DISTRIBUTION).
C                           UNITS: EV
C                           DIMENSION: TEMPERATURE INDEX.
C          (R*8)  QI()    = COLLISIONAL EXCITATION RATE COEFFICIENTS
C                           FROM IMPACT PARAMETER APPROXIMATION.
C                           UNITS: CM3 SEC-1
C                           DIMENSION: TEMPERATURE INDEX.
C          (R*8)  QJ()    = COLLISIONAL DEEXCITATION RATE COEFFICIENTS
C                           FROM IMPACT PARAMETER APPROXIMATION.
C                           UNITS: CM3 SEC-1
C                           DIMENSION: TEMPERATURE INDEX.
C          (R*8)  GA()    = GAMMA RATE PARAMETERS FROM IMPACT PARAMETER
C                           APPROXIMATION.
C                           UNITS:
C                           DIMENSION: TEMPERATURE INDEX.
C          (R*8)  XSJ()   =
C                           DIMENSION: 2
C          (R*8)  XJJ()   =
C                           DIMENSION: 2
C          (R*8)  XEJ()   =
C                           DIMENSION: 2
C          (R*8)  XSI()   =
C                           DIMENSION: 2
C          (R*8)  XJI()   =
C                           DIMENSION: 2
C          (R*8)  XEI()   =
C                           DIMENSION: 2
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4UNIT     ADAS      RETURNS UNIT NO. FOR OUTPUT OF MESSAGES.
C          I4IDFL     ADAS      RETURNS UNIQUE INDEX GIVEN QUANTUM
C                               NUMBERS N AND L.
C          R8ZETA     ADAS
C          R8WIG6     ADAS
C          CXHYDE     ADAS      CALCULATES BINDING ENERGY FOR H-LIKE
C                               ION.
C          CXLTHE     ADAS      CALCULATES BINDING ENERGY FOR LI-LIKE
C                               ION.
C          CXSODE     ADAS      CALCULATES BINDING ENERGY FOR NA-LIKE
C                               ION.
C          CXCRPS     ADAS      CALCULATES COLLISON RATE COEFFICIENTS
C                               FROM PENGELLY AND SEATON.
C          CXCRIP     ADAS      CALCULATES COLLISON RATE COEFFICIENTS
C                               FROM IMPACT PARAMETER APPROXIMATION.
C
C NOTES:
C       1) THE J->J' TRANSITION INDEX IS AS FOLLOWS:
C              1 : J=L+0.5 -> J'=L'+0.5
C              2 : J=L+0.5 -> J'=L'-0.5
C              3 : J=L-0.5 -> J'=L'+0.5
C              4 : J=L-0.5 -> J'=L'-0.5
C
C       2) THE SCALING FACTOR 'SCF' USED IN CALCULATING THE POSITIVE ION
```

```
C          RATES IS NEEDED BECAUSE THE CROSS-SECTION ROUTINES REQUIRE AN
C          INTEGER VALUE FOR THE ION CHARGE. THE ADJACENT LOWER INTEGER
C          VALUE IS USED AND THE RESULTS SCALED BY 'SCF'.
C
C AUTHOR:  JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 5183
C
C DATE:    08/10/93
C
C UPDATE:  01/11/93 - J NASH    - ADAS91:
C          UPDATED TO MORE GENERAL FORM TO ALLOW USE BY EITHER ADAS306
C          OR ADAS308. NOW HANDLES H-, LI-, OR NA-LIKE IONS, AND
C          RETURNS EXPLICIT RATES FOR J-RESOLVED TRANSITIONS.
C
C
C VERSION : 1.2
C DATE    : 28-09-2005
C MODIFIED: Martin O'Mullane
C               - Scale the gamma (GAP) in addition to the rates
C                 since 308 and 309 use this and ignore QIP and QIM.
C
C VERSION : 1.3
C DATE    : 17-05-2007
C MODIFIED: Allan Whiteford
C               - Updated comments as part of subroutine documentation
C                 procedure.
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IZ0,         IZ1,         LI,          LJ
      INTEGER             MXJSHL,      MXNSHL,      NI,          NJ
      REAL*8              DENS,        EMP,         GAE,         GAP
      REAL*8              QEM(2*MXJSHL),            QEP(2*MXJSHL)
      REAL*8              QIM(2*MXJSHL),            QIP(2*MXJSHL)
      REAL*8              TBLF((MXNSHL*(MXNSHL+1))/2),           TEV
      REAL*8              TPV,         ZP
```

4.80 cxcrip: Subroutine cxcrip from library adas3xx

```
      SUBROUTINE CXCRIP( MXCOLL , MXTEMP , IZT    , IZC    , WI     ,
     &                   EI     , WJ     , EJ     , EM     , PHI    ,
     &                   NCOLL  , EPS    , OMEG   , NTEMP  , TVA    ,
     &                   RAT    , QI     , QJ     , GA
     &                 )
C
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: CXCRIP *********************
C
C  PURPOSE:  CALCULATES ELECTRON AND POSITIVE ION COLLISION EXCITATION
C            AND DEEXCITATION RATE COEFFICIENTS FOR DIPOLE TRANSITIONS
C            IN THE IMPACT PARAMETER APPROXIMATION.
C
C            (BURGESS AND SUMMERS,1976,MON.NOT.R.AST.SOC.,174,345)
C
C            OPTIONALLY A SET OF INCIDENT PARTICLE ENERGIES AND
C            COLLISION STRENGTHS MAY BE PROVIDED, IN WHICH CASE THE
C            IMPACT PARAMETER THEORY IS USED TO CALCULATE THE COLLISION
C            STRENGTHS AT HIGH ENERGY WITH VALUES SCALED TO THE HIGHEST
C            ENERGY INPUT COLLISION STRENGTH.
C
C  CALLING PROGRAM:  CXCRDG
C
C  INPUT : (I*4)  MXCOLL  = MAXIMUM NUMBER OF COLLISION STRENGTHS.
C  INPUT : (I*4)  MXTEMP  = MAXIMUM NUMBER OF TEMPERATURES.
C  INPUT : (I*4)  IZT     = TARGET ION CHARGE.
C  INPUT : (I*4)  IZC     = CHARGE OF COLLIDING PARTICLE.
C  INPUT : (R*8)  WI      = STATISTICAL WEIGHT OF STATE I.
C  INPUT : (R*8)  EI      = BINDING ENERGY OF STATE I.
C                           UNITS: RYD
C  INPUT : (R*8)  WJ      = STATISTICAL WEIGHT OF STATE J.
C  INPUT : (R*8)  EJ      = BINDING ENERGY OF STATE J.
C                           UNITS: RYD
C  INPUT : (R*8)  EM      = REDUCED MASS FOR COLLIDING PARTICLE.
C                           UNITS: ELECTRON MASSES
C  INPUT : (R*8)  PHI     = FIJ/EIJ WHERE:
C                           FIJ = ABSORPTION OSCILLATOR STRENGTH;
C                           EIJ = EI-EJ = THE TRANSITION ENERGY (RYD).
C  INPUT : (I*4)  NCOLL   = NUMBER OF TABULAR VALUES OF COLLISION
C                           STRENGTH.
C  INPUT : (R*8)  EPS()   = INCIDENT ELECTRON ENERGIES.
C                           UNITS: RYD
C                           DIMENSION: COLLISION INDEX.
C  INPUT : (R*8)  OMEG()  = COLLISION STRENGTHS.
C                           DIMENSION: COLLISION INDEX.
C  INPUT : (I*4)  NTEMP   = NUMBER OF TEMPERATURES.
C  INPUT : (R*8)  TVA()   = TEMPERATURES (INCIDENT PARTICLE
C                           DISTRIBUTION).
C                           UNITS: EV.
C                           DIMENSION: TEMPERATURE INDEX.
C
C  OUTPUT: (R*8)  RAT     = RATIO OF OMEG(NCOLL) TO I.P. OMEGA.
C  OUTPUT: (R*8)  QI()    = COLLISIONAL EXCITATION RATE COEFFICIENTS.
C                           UNITS: CM3 SEC-1
C                           DIMENSION: TEMPERATURE INDEX.
C  OUTPUT: (R*8)  QJ()    = COLLISIONAL DEEXCITATION RATE COEFFICIENTS.
C                           UNITS: CM3 SEC-1
```

```
C                             DIMENSION: TEMPERATURE INDEX.
C  OUTPUT: (R*8)  GA()     = GAMMA RATE PARAMETERS.
C                             UNITS:
C                             DIMENSION: TEMPERATURE INDEX.
C
C          (I*4)  I        = LOOP INDEX.
C          (I*4)  K        = LOOP INDEX.
C
C          (R*8)  Z1       = IZT+1.
C          (R*8)  Z2       = ABS( IZC ).
C          (R*8)  EIJ      = TRANSITION ENERGY.
C                             UNITS: RYD
C          (R*8)  ELAM     =
C          (R*8)  ZCOL     =
C          (R*8)  Z2PHI    =
C          (R*8)  SC       =
C          (R*8)  T1       =
C          (R*8)  T2       =
C          (R*8)  EPSM     =
C          (R*8)  R        =
C          (R*8)  EIQ      =
C          (R*8)  FLAG     =
C          (R*8)  ATE      =
C          (R*8)  S        =
C          (R*8)  U1       =
C          (R*8)  G1       =
C          (R*8)  U2       =
C          (R*8)  G2       =
C          (R*8)  B        =
C          (R*8)  S1       =
C          (R*8)  E        =
C          (R*8)  P        =
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          CXEIQP     ADAS
C
C AUTHOR:  JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 5183
C
C DATE:    08/10/93
C
C VERSION: 1.1                          DATE: 20-06-95
C MODIFIED: TIM HAMMOND (Probably)
C               - PUT UNDER S.C.C.S. CONTROL
C
C VERSION  : 1.2                        DATE: 17-05-07
C MODIFIED : Allan Whiteford
C               - Updated comments as part of subroutine documentation
C                 procedure.
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IZC,          IZT,          MXCOLL,      MXTEMP
      INTEGER             NCOLL,        NTEMP
      REAL*8              EI,           EJ,           EM
      REAL*8              EPS(MXCOLL),  GA(MXTEMP),   OMEG(MXCOLL)
      REAL*8              PHI,          QI(MXTEMP),   QJ(MXTEMP),  RAT
      REAL*8              TVA(MXTEMP),  WI,           WJ
```

4.81 cxcrps: Subroutine cxcrps from library adas3xx

```
      SUBROUTINE CXCRPS( IZT  , IZC  , N    , LI   , LJ   ,
     &                   WI   , EI   , WJ   , EJ   , EM   ,
     &                   PHI  , TV   , TEV  , DENS , TAU  ,
     &                   QI   , QJ   , GA
     &                 )
C
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: CXCRIP *********************
C
C  PURPOSE:  CALCULATES PENGELLY & SEATON (1964) COLLISION RATES BETWEEN
C            NEARLY DEGENERATE LEVELS. A VARIATION OF IMPACT PARAMETER
C            THEORY FOR DIPOLE TRANSITIONS IS USED.
C
C  CALLING PROGRAM:  CXCRDG
C
C  INPUT : (I*4)  IZT     = TARGET ION CHARGE.
C  INPUT : (I*4)  IZC     = CHARGE OF COLLIDING PARTICLE.
C  INPUT : (I*4)  N       = PRINCIPAL QUANTUM NUMBER.
C  INPUT : (I*4)  LI      = ORBITAL QUANTUM NUMBER.
C  INPUT : (I*4)  LJ      = ORBITAL QUANTUM NUMBER.
C  INPUT : (R*8)  WI      = STATISTICAL WEIGHT OF STATE I.
C  INPUT : (R*8)  EI      = BINDING ENERGY OF STATE I.
C                           UNITS: RYD
C  INPUT : (R*8)  WJ      = STATISTICAL WEIGHT OF STATE J.
C  INPUT : (R*8)  EJ      = BINDING ENERGY OF STATE J.
C                           UNITS: RYD
C  INPUT : (R*8)  EM      = REDUCED MASS FOR COLLIDING PARTICLE.
C                           UNITS: ELECTRON MASSES
C  INPUT : (R*8)  PHI     = FIJ/EIJ WHERE:
C                           FIJ = ABSORPTION OSCILLATOR STRENGTH;
C                           EIJ = EI-EJ = THE TRANSITION ENERGY (RYD).
C  INPUT : (R*8)  TV      = TEMPERATURE (COLLIDING PARTICLE
C                           DISTRIBUTION).
C                           UNITS: EV
C  INPUT : (R*8)  TEV     = TEMPERATURE (ELECTRON DISTRIBUTION).
C                           UNITS: EV
C  INPUT : (R*8)  DENS    = ELECTRON DENSITY.
C                           UNITS: CM-3
C  INPUT : (R*8)  TAU     = MEAN RADIATIVE LIFETIME OF INTIAL AND FINAL
C                           LEVELS.
C                           UNITS: SEC
C
C  OUTPUT: (R*8)  QI      = EXCITATION RATE COEFFICIENT.
C                           UNITS: CM3 SEC-1
C  OUTPUT: (R*8)  QJ      = DEEXCITATION RATE COEFFICIENTS.
C                           UNITS: CM3 SEC-1
C  OUTPUT: (R*8)  GA      = GAMMA RATE PARAMETER.
C                           UNITS:
C
C  PARAM : (R*8)  P1      =
C  PARAM : (R*8)  P2      =
C
C          (I*4)  IND1    =
C                           : 0 = FINITE RADIATIVE LIFETIME CUT-OFF.
C                           : 1 = BETHE CUT-OFF.
C          (I*4)  IND2    =
C                           : 0 = LIFETIME OR BETHE CUT-OFF.
```

```
C                           : 1 = DEBYE CUT-OFF.
C
C          (R*8)  T       = TEMPERATURE (COLLIDING PARTICLE
C                           DISTRIBUTION).
C                           UNITS:
C          (R*8)  TE      = TEMPERATURE (ELECTRON DISTRIBUTION).
C                           UNITS:
C          (R*8)  ATP     =
C          (R*8)  Z1      = ZT+1.
C          (R*8)  XN      = REAL VALUE = N.
C          (R*8)  XLI     = REAL VALUE = LI.
C          (R*8)  XLJ     = REAL VALUE = LJ.
C          (R*8)  XL      =
C          (R*8)  DNL     =
C          (R*8)  EIJ     =
C          (R*8)  TAU1    =
C          (R*8)  F1      =
C          (R*8)  F       =
C          (R*8)  B       =
C
C ROUTINES: NONE
C
C AUTHOR:  JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 5183
C
C DATE:    11/10/93
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IZC,         IZT,         LI,          LJ
      INTEGER             N
      REAL*8              DENS,        EI,          EJ,          EM
      REAL*8              GA,          PHI,         QI,          QJ
      REAL*8              TAU,         TEV,         TV,          WI
      REAL*8              WJ
```

4.82 cxdata: Subroutine cxdata from library adas3xx

```
       SUBROUTINE CXDATA( IUNIT  , MXNENG , MXNSHL , TITLED ,
     &                    SYMBR  , SYMBD  , IZR    , IZD    ,
     &                    INDD   , NENRGY , NMIN   , NMAX   ,
     &                    LPARMS , LSETL  , LSETM  , ENRGYA ,
     &                    ALPHAA , LFORMA , XLCUTA , PL2A   ,
     &                    PL3A   , SIGTA  , SIGNA  , SIGLA  ,
     &                    SIGMA
     &                  )
C
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: CXDATA *********************
C
C  PURPOSE:  TO FETCH DATA FROM INPUT DATA SET OF TYPE ADF01.
C
C  CALLING PROGRAM: ADAS301/ADAS306/ADAS307/ADAS308/ADAS309
C
C  DATA:
C
C           THE UNITS USED IN THE DATA FILE ARE TAKEN AS FOLLOWS:
C
C           COLLISION ENERGIES  : KEV/AMU
C           ALPHA               :
C           TOTAL XSECTS.       : CM2
C           N-SHELL XSECTS.     : CM2
C           NL-SHELL DATA       : CM2
C           NLM-SHELL DATA      : CM2
C
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  IUNIT     = UNIT TO WHICH INPUT FILE IS ALLOCATED.
C  INPUT : (I*4)  MXNENG    = MAXIMUM NO. OF ENERGIES.
C  INPUT : (I*4)  MXNSHL    = MAXIMUM NO. OF N SHELLS.
C
C  OUTPUT: (C*80) TITLED    = NOT SET - TITLE FOR DATA SOURCE.
C  OUTPUT: (C*2)  SYMBR     = READ - RECEIVER ION ELEMENT SYMBOL.
C  OUTPUT: (C*2)  SYMBD     = READ - DONOR ION ELMENT SYMBOL.
C  OUTPUT: (I*4)  IZR       = READ - ION CHARGE OF RECEIVER.
C  OUTPUT: (I*4)  IZD       = READ - ION CHARGE OF DONOR.
C  OUTPUT: (I*4)  INDD      = READ - DONOR STATE INDEX.
C  OUTPUT: (I*4)  NENRGY    = NUMBER OF ENERGIES READ.
C  OUTPUT: (I*4)  NMIN      = LOWEST N-SHELL FOR WHICH DATA READ.
C  OUTPUT: (I*4)  NMAX      = HIGHEST N-SHELL FOR WHICH DATA READ.
C  OUTPUT: (L*4)  LPARMS    = FLAGS IF L-SPLITTING PARAMETERS PRESENT.
C                             .TRUE.  => L-SPLITTING PARAMETERS PRESENT.
C                             .FALSE  => L-SPLITTING PARAMETERS ABSENT.
C  OUTPUT: (L*4)  LSETL     = FLAGS IF L-RESOLVED DATA PRESENT.
C                             .TRUE.  => L-RESOLVED DATA PRESENT.
C                             .FALSE  => L-RESOLVED DATA ABSENT.
C  OUTPUT: (L*4)  LSETM     = FLAGS IF M-RESOLVED DATA PRESENT.
C                             .TRUE.  => M-RESOLVED DATA PRESENT.
C                             .FALSE  => M-RESOLVED DATA ABSENT.
C
C  OUTPUT: (R*8)  ENRGYA()  = READ - COLLISION ENERGIES.
C                             UNITS: EV/AMU (READ AS KEV/AMU)
C                             DIMENSION: ENERGY INDEX
C  OUTPUT: (R*8)  ALPHAA()  = READ - EXTRAPOLATION PARAMETER ALPHA.
```

```
C                                DIMENSION: ENERGY INDEX
C  OUTPUT: (I*4)  LFORMA()  = READ - PARAMETERS FOR CALCULATING L-RES
C                                       X-SEC.
C                                DIMENSION: ENERGY INDEX
C  OUTPUT: (R*8)  XLCUTA()  = READ - PARAMETERS FOR CALCULATING L-RES
C                                       X-SEC.
C                                DIMENSION: ENERGY INDEX
C  OUTPUT: (R*8)  PL2A()    = READ - PARAMETERS FOR CALCULATING L-RES
C                                       X-SEC.
C                                DIMENSION: ENERGY INDEX
C  OUTPUT: (R*8)  PL3A()    = READ - PARAMETERS FOR CALCULATING L-RES
C                                       X-SEC.
C                                DIMENSION: ENERGY INDEX
C  OUTPUT: (R*8)  SIGTA()   = READ - TOTAL CHARGE EXCHANGE
C                                       CROSS-SECTION.
C                                UNITS: CM2
C                                DIMENSION: ENERGY INDEX
C  OUTPUT: (R*8)  SIGNA(,)  = READ - N-RESOLVED CHARGE EXCHANGE
C                                       CROSS-SECTIONS.
C                                UNITS: CM2
C                                1ST DIMENSION: ENERGY INDEX
C                                2ND DIMENSION: N-SHELL
C  OUTPUT: (R*8)  SIGLA(,)  = READ - L-RESOLVED CHARGE EXCHANGE
C                                       CROSS-SECTIONS.
C                                UNITS: CM2
C                                1ST DIMENSION: ENERGY INDEX
C                                2ND DIMENSION: INDEXED BY I4IDFL(N,L)
C  OUTPUT: (R*8)  SIGMA(,)  = READ - M-RESOLVED CHARGE EXCHANGE
C                                       CROSS-SECTIONS.
C                                UNITS: CM2
C                                1ST DIMENSION: ENERGY INDEX
C                                2ND DIMENSION: INDEXED BY I4IDFM(N,L,M)
C                                               WITH M >= 0 ONLY
C
C    (R*8)  ZEROST   = PARAMETER = EFFECTIVE SHIFT APPLIED TO
C  CROSS-SECTION VALUES TO AVOID
C  ZERO VALUES (WILL NOT AFFECT
C  ANY VALUES WHICH ARE GREATER
C  THAN AROUND 1.0E+15*ZEROSHFT -
C  i.e. 1.0E-25.)
C
C          (I*4)  OLDMIN   = PREVIOUS VALUE READ FOR NMIN.
C          (I*4)  OLDMAX   = PREVIOUS VALUE READ FOR NMAX.
C          (I*4)  IBLK     = CURRENT DATA BLOCK.
C          (I*4)  IVALUE   = USED TO PARSE FOR END OF DATA FLAG (-1).
C          (I*4)  N        = N QUANTUM NUMBER.
C          (I*4)  L        = L QUANTUM NUMBER.
C          (I*4)  M        = M QUANTUM NUMBER.
C          (I*4)  I        = LOOP COUNTER.
C          (I*4)  J        = LOOP COUNTER.
C          (I*4)  IERR     = ERROR RETURN CODE.
C          (C*2)  CIZR     = ION CHARGE OF RECEIVER.
C          (C*2)  CIZD     = ION CHARGE OF DONOR.
C          (C*1)  INDD     = DONOR STATE INDEX.
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4FCTN     ADAS      RETURNS CHARACTER STRING AS AN INTEGER.
C          I4UNIT     ADAS      FETCH UNIT NUMBER FOR OUTPUT OF MESSAGES
C          I4IDFL     ADAS      RETURNS UNIQUE INDEX FROM QUANTUM
```

```
C                               NUMBERS N AND L.
C          I4IDFM     ADAS      RETURNS UNIQUE INDEX FROM QUANTUM
C                               NUMBERS N, L AND M.
C          XXIDTL     ADAS      INVERSE OF I4IDFL. RETURNS QUANTUM
C                               NUMBERS N AND L FROM INDEX.
C          XXIDTM     ADAS      INVERSE OF I4IDFM. RETURNS QUANTUM
C                               NUMBERS N, L AND M FROM INDEX.
C
C AUTHOR:  JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 5183
C
C DATE:    21/09/93
C
C UPDATE:  18/10/93 - J NASH    - ADAS91:
C          UPDATED TO READ L-SPLITTING PARAMETERS IF PRESENT IN DATASET.
C
C UPDATE:  01/05/95 - Tim Hammond - IDLADAS:
C    UNIX port.
C
C UPDATE:  16/05/95 - Tim Hammond - IDLADAS:
C    ADDED AND APPLIED ZEROST PARAMETER => EFFECTIVE ZERO FOR
C          CROSS-SECTIONS (CODING DONE BY PAUL BRIDEN).
C
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      CHARACTER*2         SYMBD,       SYMBR
      CHARACTER*80        TITLED
      INTEGER             INDD,        IUNIT,       IZD,          IZR
      INTEGER             LFORMA(MXNENG),           MXNENG,       MXNSHL
      INTEGER             NENRGY,      NMAX,        NMIN
      LOGICAL             LPARMS,      LSETL,       LSETM
      REAL*8              ALPHAA(MXNENG),           ENRGYA(MXNENG)
      REAL*8              PL2A(MXNENG),             PL3A(MXNENG)
      REAL*8              SIGLA(MXNENG,(MXNSHL*(MXNSHL+1))/2)
      REAL*8              SIGMA(MXNENG,(MXNSHL*(MXNSHL+1)*(MXNSHL+2))/6)
      REAL*8              SIGNA(MXNENG,MXNSHL),     SIGTA(MXNENG)
      REAL*8              XLCUTA(MXNENG)
```

4.83 cxextr: Subroutine cxextr from library adas3xx

```
      SUBROUTINE CXEXTR( MXNENG , MXNSHL , NMINF  , NMAXF  ,
     &                   NENRGY , LPARMS , ALPHAA , LFORMA ,
     &                   XLCUTA , PL2A   , PL3A   , XSECNA ,
     &                   FRACLA
     &                 )
C
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: CXEXTR *********************
C
C  PURPOSE:  EXTRAPOLATES N AND L RESOLVED CROSS-SECTION BELOW AND ABOVE
C            DATA IN INPUT DATASET.
C
C            N AND L RESOLVED CROSS-SECTIONS BELOW THE INPUT DATA ARE
C            SET TO ZERO.
C
C            N RESOLVED CROSS-SECTIONS ABOVE THE INPUT DATA ARE
C            CALCULATED USING THE EXTRAPOLATION EXPONENT 'ALPHA'.
C
C            L-RESOLVED CROSS-SECTIONS ABOVE THE INPUT DATA ARE
C            CALCULATED USING THE FUNCTION R8FORM IF THE L-SPLITTING
C            PARAMTERS ARE PRESENT, OTHERWISE THE DISTRIBUTION FROM THE
C            INPUT DATA FOR THE MAXIMUM PRINCIPAL QUANTUM NUMBER IS
C            COPIED TO THE HIGHER LEVELS AS FOLLOWS:
C
C            SIGMA(N,L) = SIGMA(NMAXF,L)   FOR L = 0 TO NMAXF-1
C                       = 0                FOR L = NMAXF TO N-1
C
C            WHERE NMAXF IS THE MAXIMUM PRINCIPAL QUANTUM NUMBER IN THE
C            INPUT DATA.
C
C  CALLING PROGRAM: ADAS308
C
C  INPUT : (I*4)  MXNENG    = MAXIMUM NO. OF ENERGIES.
C  INPUT : (I*4)  MXNSHL    = MAXIMUM NO. OF N SHELLS.
C  INPUT : (I*4)  NMINF     = LOWEST N-SHELL FOR WHICH DATA READ.
C  INPUT : (I*4)  NMAXF     = HIGHEST N-SHELL FOR WHICH DATA READ.
C  INPUT : (I*4)  NENRGY    = NUMBER OF ENERGIES IN DATASET.
C  INPUT : (L*4)  LPARMS    = FLAGS IF L-SPLITTING PARAMETERS PRESENT.
C                             .TRUE.  => L-SPLITTING PARAMETERS PRESENT.
C                             .FALSE  => L-SPLITTING PARAMETERS ABSENT.
C  INPUT : (R*8)  ALPHAA()  = EXTRAPOLATION PARAMETER ALPHA.
C                             DIMENSION: ENERGY INDEX
C  INPUT : (I*4)  LFORMA()  = PARAMETERS FOR CALCULATING L-RES X-SEC.
C                             DIMENSION: ENERGY INDEX
C  INPUT : (R*8)  XLCUTA()  = PARAMETERS FOR CALCULATING L-RES X-SEC.
C                             DIMENSION: ENERGY INDEX
C  INPUT : (R*8)  PL2A()    = PARAMETERS FOR CALCULATING L-RES X-SEC.
C                             DIMENSION: ENERGY INDEX
C  INPUT : (R*8)  PL3A()    = PARAMETERS FOR CALCULATING L-RES X-SEC.
C                             DIMENSION: ENERGY INDEX
C
C  I/O   : (R*8)  XSECNA(,) = N-RESOLVED CHARGE EXCHANGE CROSS-SECTIONS.
C                             UNITS: CM2
C                             1ST DIMENSION: ENERGY INDEX
C                             2ND DIMENSION: N-SHELL
C  I/O   : (R*8)  FRACLA(,) = L-RESOLVED CHARGE EXCHANGE CROSS-SECTIONS
C                             EXPRESSED AS FRACTION OF CORRESPONDING
```

```
C                             N-RESOLVED CROSS-SECTION.
C                             1ST DIMENSION: ENERGY INDEX
C                             2ND DIMENSION: REFERENCED BY I4IDFL(N,L)
C
C  PARAM : (R*8)  ALPMIN    = MINIMUM VLUE OF EXTRPOLATION EXPONENT
C                             'ALPHA'.
C
C          (I*4)  IE         = ENERGY INDEX.
C          (I*4)  N          = PRINCIPAL QUANTUM NUMBER.
C          (I*4)  L          = ORBITAL QUANTUM NUMBER.
C          (I*4)  IDL        = L-RESOLVED DATA INDEX.
C          (I*4)  IDL1       = L-RESOLVED DATA INDEX.
C          (I*4)  IDL2       = L-RESOLVED DATA INDEX.
C          (I*4)  ITYPE      = TYPE OF APPROX. TO USE IN FUNC R8FORM.
C
C          (R*8)  XNMAXF     = REAL VALUE = NMAXF.
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4IDFL     ADAS      RETURNS UNIQUE INDEX FROM QUANTUM
C                               NUMBERS N AND L.
C          R8FORM     ADAS      RETURNS L-RES X-SEC AS FRACTION OF N-RES
C                               X-SEC.
C
C AUTHOR:  JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 5183
C
C DATE:    20/10/93
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             LFORMA(MXNENG),           MXNENG,      MXNSHL
      INTEGER             NENRGY,      NMAXF,       NMINF
      LOGICAL             LPARMS
      REAL*8              ALPHAA(MXNENG)
      REAL*8              FRACLA(MXNENG,(MXNSHL*(MXNSHL+1))/2)
      REAL*8              PL2A(MXNENG),             PL3A(MXNENG)
      REAL*8              XLCUTA(MXNENG)
      REAL*8              XSECNA(MXNENG,MXNSHL)
```

4.84 cxfrac: Subroutine cxfrac from library adas3xx

```
       SUBROUTINE CXFRAC( MXNENG , MXNSHL ,
     &                    NENRGY , NMIN   , NMAX   ,
     &                    LSETL  ,
     &                    SIGNA  , FRACLA
     &                  )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: CXFRAC *********************
C
C  PURPOSE:  TO CONVERT L RESOLVED PARTIAL CROSS-SECTIONS FROM
C            ABSOLUTE VALUES TO FRACTIONS OF THE N AND L RESOLVED DATA,
C            RESPECTIVELY.
C
C  CALLING PROGRAM: ADAS301/ADAS306/ADAS307/ADAS308/ADAS309
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  MXNENG    = MAXIMUM NO. OF ENERGIES.
C  INPUT : (I*4)  MXNSHL    = MAXIMUM NO. OF N SHELLS.
C  INPUT : (I*4)  NENRGY    = NUMBER OF ENERGIES READ.
C  INPUT : (I*4)  NMIN      = LOWEST N-SHELL FOR WHICH DATA READ.
C  INPUT : (I*4)  NMAX      = HIGHEST N-SHELL FOR WHICH DATA READ.
C  INPUT : (L*4)  LSETL     = FLAGS IF L-RESOLVED DATA PRESENT.
C                             .TRUE.  => L-RESOLVED DATA PRESENT.
C                             .FALSE  => L-RESOLVED DATA ABSENT.
C  INPUT : (R*8)  SIGNA(,)  = N-RESOLVED CHARGE EXCHANGE CROSS-SECTIONS.
C                             UNITS: CM2
C                             1ST DIMENSION: ENERGY INDEX
C                             2ND DIMENSION: N-SHELL
C
C  I/O   : (R*8)  FRACLA(,) = L-RESOLVED CHARGE EXCHANGE CROSS-SECTIONS.
C                             INPUT : ABSOLUTE VALUES (CM2)
C                             OUTPUT: FRACTION OF N-RESOLVED DATA.
C                             1ST DIMENSION: ENERGY INDEX
C                             2ND DIMENSION: INDEXED BY I4IDFL(N,L)
C
C          (I*4)  IE        = ENERGY INDEX.
C          (I*4)  N         = N QUANTUM NUMBER.
C          (I*4)  L         = L QUANTUM NUMBER.
C          (I*4)  IDL       = L-RESOLVED DATA INDEX.
C          (I*4)  IDM       = M-RESOLVED DATA INDEX.
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4IDFL     ADAS      RETURNS UNIQUE INDEX FROM QUANTUM
C                               NUMBERS N AND L.
C          I4IDFM     ADAS      RETURNS UNIQUE INDEX FROM QUANTUM
C                               NUMBERS N, L AND M.
C
C AUTHOR:  JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 5183
C
C DATE:    15/09/93
C
C VERSION  : 1.2
C DATE     : 22-05-2007
C MODIFIED : Martin O'Mullane
```

```
C              - Remove unused m-subshell data possibility.
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      INTEGER             MXNENG,      MXNSHL,      NENRGY,      NMAX
      INTEGER             NMIN
      LOGICAL             LSETL
      REAL*8              FRACLA(MXNENG,(MXNSHL*(MXNSHL+1))/2)
      REAL*8              SIGNA(MXNENG,MXNSHL)
```

4.85 cxgfil: Subroutine cxgfil from library adas3xx

```
      SUBROUTINE CXGFIL( MXA    , MXB    , IZ0    , AMSSNO ,
     &                   N      , N1     , TIEV   , NDIV   ,
     &                   ANGDIV , NPTS   , CEMIS  , WAVLN  ,
     &                   CWLN   , XA     , YA     , XB     ,
     &                   YB
     &                 )
C
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: CXGFIL *********************
C
C  PURPOSE: FILLS ADAS306 AND 308 GRAPH ARRAYS.
C
C  CALLING PROGRAM: C6OUTG ; C8OUTG.
C
C  INPUT : (I*4)  MXA       = MAXIMUM NUMBER OF POINTS IN GRAPH A.
C  INPUT : (I*4)  MXB       = MAXIMUM NUMBER OF POINTS IN GRAPH B.
C  INPUT : (I*4)  IZ0       = TARGET NUCLEAR CHARGE.
C  INPUT : (R*8)  AMSSNO    = ATOMIC MASS NUMBER OF TARGET.
C  INPUT : (I*4)  N         = UPPER PRINCIPAL QUANTUM NUMBER OF
C                             TRANSITION.
C  INPUT : (I*4)  N1        = LOWER PRINCIPAL QUANTUM NUMBER OF
C                             TRANSITION.
C  INPUT : (R*8)  TIEV      = ION TEMPERATURE.
C                             UNITS: A
C  INPUT : (I*4)  NDIV      = NUMBER OF DIVISIONS ON X AXIS OF PLOT.
C  INPUT : (R*8)  ANGDIV    = NO. OF ANGSTROMS PER DIVISION.
C  INPUT : (I*4)  NPTS      = NUMBER OF POINTS OF DATA.
C  INPUT : (R*8)  CEMIS()   = COLUMN EMISSIVITIES.
C                             UNITS: PHOT CM-2 SEC-1
C                             DIMENSION: MXA
C  INPUT : (R*8)  WAVLN()   = WAVELENGTHS.
C                             UNITS: A
C                             DIMENSION: MXA
C
C  OUTPUT  (R*8)  CWLN      = CENTRAL WAVELENGTH ON GRAPH.
C                             UNITS: A
C  OUTPUT: (R*8)  XA()      = X DATA POINTS FOR GRAPH A.
C                             DIMENSION: MXA
C  OUTPUT: (R*8)  YA()      = Y DATA POINTS FOR GRAPH A.
C                             DIMENSION: MXA
C  OUTPUT: (R*8)  XB()      = X DATA POINTS FOR GRAPH B.
C                             DIMENSION: MXB
C  OUTPUT: (R*8)  YB()      = Y DATA POINTS FOR GRAPH B.
C                             DIMENSION: MXB
C
C          (I*4)  MIDDIV    = CENTRAL DIVISION ON GRAPH.
C          (I*4)  I         = LOOP INDEX.
C          (I*4)  J         = LOOP INDEX.
C
C          (R*8)  Z0        = REAL VALUE = IZ0.
C          (R*8)  XN        = REAL VALUE = N.
C          (R*8)  XN1       = REAL VALUE = N1.
C          (R*8)  G         =
C
C          (R*8)  MINVAL    = MINIMUM VALUE BELOW WHICH YB IS SET TO ZERO
C
C ROUTINES: NONE
```

```
C
C AUTHOR:  JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/87
C          JET EXT. 5183
C
C DATE:    22/11/93
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IZ0,         MXA,          MXB,          N
      INTEGER             N1,          NDIV,         NPTS
      REAL*8              AMSSNO,      ANGDIV,       CEMIS(MXA),   CWLN
      REAL*8              TIEV,        WAVLN(MXA),   XA(MXA),      XB(MXB)
      REAL*8              YA(MXA),     YB(MXB)
```

4.86 cxghnl: Subroutine cxghnl from library adas3xx

```
      SUBROUTINE CXGHNL ( MXNSHL , IZ1    , IZEFF  , N      ,
     &                    L      , N1     , TEV    , GAMA
     &                  )
C
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: CXGHNL *********************
C
C  PURPOSE:  CALCULATES APPROXIMATE EXCITATION RATE PARAMETERS FROM
C            N,L LEVELS OF HYDROGEN-LIKE AND LITHIUM-LIKE IONS TO HIGHER
C            LEVELS N1,L1 USING CLASSICAL OVERLAPS.
C
C            FOR TRANSITIONS FROM 1S,2S AND 2P APPROXIMATE FITTINGS ARE
C            USED BASED ON SAMPSON DATA.
C
C            FOR TRANSITIONS FROM HIGHER NL LEVELS UPWARDS A RESOLVED
C            VARIANT OF PERCIVAL-RICHARDS IS USED, WITH NUMERICAL
C            QUADRATURES.
C
C  CALLING PROGRAM: ADAS308.
C
C  INPUT : (I*4)  MXNSHL   = MAXIMUM VALUE OF N QUANTUM NUMBER.
C  INPUT : (I*4)  IZ1      = ION CHARGE.OF ION
C  INPUT : (R*8)  IZEFF    = EFFECTIVE ION CHARGE (CF. SAMPSON ET AL.).
C  INPUT : (I*4)  N        = LOWER VALUE OF N QUANTUM NUMBER.
C  INPUT : (I*4)  L        = L QUANTUM NUMBER FOR NL.
C  INPUT : (I*4)  N1       = UPPER VALUE OF N QUANTUM NUMBER.
C  INPUT : (R*8)  TEV      = ELECTRON TEMPERATURE.
C                            UNITS: EV
C
C  OUTPUT: (R*8)  GAMA()   = TABLE OF RATE PARAMETERS.
C                            UNITS:
C                            DIMENSION: REFERENCED BY N QUANTUM NUMBER.
C
C  PARAM : (I*4)  MXN      = 'MXNSHL'.
C  PARAM : (I*4)  NMAX     =
C  PARAM : (R*8)  P1       =
C  PARAM : (R*8)  P2       =
C  PARAM : (R*8)  C1       = 2/3.
C  PARAM : (R*8)  C2       = 4/3.
C
C          (I*4)  L1       = L QUANTUM NUMBER FOR N1.
C          (I*4)  K        =
C          (I*4)  I        = LOOP INDEX.
C          (I*4)  J        = LOOP INDEX.
C
C          (R*8)  Z1       = REAL VALUE = IZ1.
C          (R*8)  ZEFF     = REAL VALUE = IZEFF.
C          (R*8)  Z12      = Z1**2.
C          (R*8)  XN       = REAL VALUE = N.
C          (R*8)  XN1      = REAL VALUE = N1.
C          (R*8)  XND      = REAL VALUE = N1-N.
C          (R*8)  XNP      = REAL VALUE = N1*N.
C          (R*8)  DE       =
C          (R*8)  ATE      =
C          (R*8)  Z2OMT    =
C          (R*8)  SUM      = SUM OF 'FOMA' TERMS.
C          (R*8)  W1       =
```

```
C          (R*8)  OVL      =
C          (R*8)  XLN      =
C          (R*8)  XLG      =
C          (R*8)  Z2S      =
C          (R*8)  GAM      =
C          (R*8)  XL       =
C          (R*8)  XM       =
C          (R*8)  XP       =
C          (R*8)  CM       =
C          (R*8)  CP       =
C          (R*8)  D        =
C          (R*8)  E        =
C          (R*8)  F        =
C          (R*8)  G        =
C          (R*8)  H        =
C          (R*8)  T        =
C          (R*8)  Y        =
C
C          (R*8)  XA()     =
C          (R*8)  WA()     =
C          (R*8)  FOMA()   =
C
C          (R*8)  FA(,)    =
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4UNIT     ADAS      RETURNS UNIT NO. FOR OUTPUT MESSAGES.
C          CXOVLP     ADAS
C          R8RD2B     ADAS      RETURNS HYDRONIC BOUND-BOUND RADIAL
C                               INTEGRALS.
C          R8FEEI     ADAS
C
C
C AUTHOR:   JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C           K1/0/81
C           JET EXT. 5183
C
C DATE:     05/10/93
C
C VERSION  : 1.2
C DATE     : 11-04-2007
C MODIFIED : Allan Whiteford
C               - Renamed R8OVLP to CXOVLP.
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IZ1,          IZEFF,       L,           MXNSHL
      INTEGER             N,            N1
      REAL*8              GAMA(MXNSHL),              TEV
```

4.87 cxhyde: Subroutine cxhyde from library adas3xx

```
      SUBROUTINE CXHYDE( IZ0 , ZEFF , N , L , E0 )
C
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: CXHYDE *********************
C
C  PURPOSE:  CALCULATES LOWEST ORDER NON-RELATIVISTIC, RELATIVISTIC AND
C            QUANTUMELECTRODYNAMIC ENERGIES FOR HYDROGENIC IONS.
C
C            BINDING ENERGY FOR CENTRE OF TERM IS PRODUCED.
C
C            FINE STRUCTURE FOR L>0 MUST BE ADDED EXTERNALLY.
C            FORMULAE ARE FROM ERIKSON (1977) J.PHY CHEM.REF.DATA,6,831.
C
C            QED EFFECTS FOR L>0 OMITTED.
C
C  CALLING PROGRAM: GENERAL USE
C
C  INPUT : (I*4)  IZ0     = NUCLEAR CHARGE.
C  INPUT : (R*8)  ZEFF    = EFFECTIVE NUCLEAR CHARGE.
C  INPUT : (I*4)  N       = PRINCIPAL QUANTUM NUMBER.
C  INPUT : (I*4)  L       = ORBITAL QUANTUM NUMBER.
C
C  OUTPUT: (R*8)  E0      = CENTER BINDING ENERGY.
C                           UNITS: RYD
C
C          (R*8)  Z0      = REAL VALUE = IZ0.
C          (R*8)  XN      = REAL VALUE = N.
C          (R*8)  XL      = REAL VALUE = L.
C          (R*8)  BLN     =
C          (R*8)  RMC     =
C          (R*8)  QED     =
C
C          (R*8)  BLNA()  =
C                           DIMENSION: 4
C
C ROUTINES: NONE
C
C AUTHOR:  JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 5183
C
C DATE:    08/10/93
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IZ0,         L,           N
      REAL*8              E0,          ZEFF
```

4.88 cxlthe: Subroutine cxlthe from library adas3xx

```
      SUBROUTINE CXLTHE( IZ0 , ZEFF , N , L , E0 )
C
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: CXLTHE *********************
C
C  PURPOSE:  PROVIDES BINDING ENERGY OF TERM CENTRE FOR OUTER ELECTRON
C            IN LITHIUM LIKE IONS.
C
C            FROM EDLEN (1979) PHYSICA SCRIPTA, 19, 255.
C
C            FINE STRUCTURE FOR L>0 MUST BE ADDED EXTERNALLY.
C
C  CALLING PROGRAM: GENERAL USE
C
C  INPUT : (I*4)  IZ0     = NUCLEAR CHARGE.
C  INPUT : (R*8)  ZEFF    = EFFECTIVE NUCLEAR CHARGE.
C  INPUT : (I*4)  N       = PRINCIPAL QUANTUM NUMBER.
C  INPUT : (I*4)  L       = ORBITAL QUANTUM NUMBER.
C
C  OUTPUT: (R*8)  E0      = BINDING ENERGY.
C                           UNITS: RYD
C
C          (R*8)  Z       = DUMMY ARGUMENT TO STATEMENT FUNCTIONS.
C          (R*8)  F       = DUMMY ARGUMENT TO STATEMENT FUNCTIONS.
C
C          (R*8)  Z0      = REAL VALUE = IZ0.
C          (R*8)  XN      = REAL VALUE = N.
C          (R*8)  XL      = REAL VALUE = L.
C          (R*8)  R       =
C          (R*8)  RC      =
C          (R*8)  T1      =
C          (R*8)  T2      =
C          (R*8)  T3      =
C          (R*8)  T4      =
C          (R*8)  A       =
C          (R*8)  B       =
C          (R*8)  C       =
C          (R*8)  T2S     =
C          (R*8)  D2S     =
C          (R*8)  T2P1    =
C          (R*8)  T2P2    =
C          (R*8)  T2P     =
C          (R*8)  D2P     =
C          (R*8)  U       =
C          (R*8)  U0      =
C          (R*8)  V       =
C          (R*8)  TNL     =
C          (R*8)  S       =
C          (R*8)  AK      =
C          (R*8)  DP      =
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4UNIT     ADAS      RETURNS UNIT NO. FOR OUTPUT OF MESSAGES.
C          R8P        ADAS
C          R8QP       ADAS
```

```
C          EL0        INTERNAL
C          SIG0       INTERNAL
C          DRS        INTERNAL
C          DRP        INTERNAL
C          DLS        INTERNAL
C          DLP        INTERNAL
C
C AUTHOR:  JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 5183
C
C DATE:    02/11/93
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IZ0,         L,           N
      REAL*8              E0,          ZEFF
```

4.89 cxmrdg: Subroutine cxmrdg from library adas3xx

```
      SUBROUTINE CXMRDG ( MXNSHL , MXJSHL , IZ0    , IZ1    ,
     &                    AMSSNO , NI     , LI     , NJ     ,
     &                    LJ     , BMAG   , TIEV   , TBLF   ,
     &                    FMP    , FMM    , FMI    , FMJ
     &                  )
C
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: CXMRDG *********************
C
C  PURPOSE:  CALCULATES MAGNETIC FIELD DEPENDENT MIXING RATE
C            COEFFICIENTS BETWEEN NEARLY DEGENERATE LEVELS FOR
C            HYDROGEN-LIKE, LITHIUM-LIKE AND SODIUM-LIKE IONS.
C
C            RATES ARE CALCULATED FOR THE SEPARATE
C            NLJ->NL+1J', NLJ->NLJ' AND NLJ->NL-1J'
C            TRANSITIONS.
C
C  CALLING PROGRAM: C6TBFM
C
C  INPUT : (I*4)  MXNSHL  = MAXIMUM NUMBER OF N SHELLS.
C  INPUT : (I*4)  MXJSHL  = MAXIMUM NUMBER OF J SUB-SHELLS.
C  INPUT : (I*4)  IZ0     = TARGET NUCLEAR CHARGE.
C  INPUT : (I*4)  IZ1     = ION CHARGE.
C  INPUT : (R*8)  AMSSNO  = ATOMIC MASS NO.
C  INPUT : (I*4)  NI      = VALENCE ELECTRON PRINCIPAL QUANTUM NUMBER
C                           IN STATE I.
C  INPUT : (I*4)  LI      = VALENCE ELECTRON ORBITAL QUANTUM NUMBER IN
C                           STATE I.
C  INPUT : (I*4)  NJ      = VALENCE ELECTRON PRINCIPAL QUANTUM NUMBER
C                           IN STATE J.
C  INPUT : (I*4)  LJ      = VALENCE ELECTRON ORBITAL QUANTUM NUMBER IN
C                           STATE J.
C  INPUT : (R*8)  BMAG    = MAGNETIC INDUCTION.
C                           UNITS: TESLA
C  INPUT : (R*8)  TIEV    = TEMPERATURE (ION DISTRIBUTION).
C                           UNITS: EV
C  INPUT : (R*8)  TBLF()  = TABLE OF RADIATIVE LIFETIMES.
C                           UNITS: SECS
C                           DIMENSION: REFERENCED BY FUNC I4IDFL(N,L).
C
C  OUTPUT: (R*8)  FMP()   = RATE COEFFT. FOR NLJ->NL+1J'.
C                           DIMENSION: J->J' TRANSITION INDEX.
C  OUTPUT: (R*8)  FMM()   = RATE COEFFT. FOR NLJ->NL+1J'.
C                           DIMENSION: J->J' TRANSITION INDEX.
C  OUTPUT: (R*8)  FMI()   = RATE COEFFT. FOR NLJ->NLJ' FOR STATE I.
C                           DIMENSION: J->J' TRANSITION INDEX.
C  OUTPUT: (R*8)  FMJ()   = RATE COEFFT. FOR NLJ->NLJ' FOR STATE J.
C                           DIMENSION: J->J' TRANSITION INDEX.
C
C  PARAM : (I*4)  MXTERM  = 2.
C  PARAM : (R*8)  P1      =
C
C          (I*4)  ICI     =
C          (I*4)  ICJ     =
C          (I*4)  I       = LOOP INDEX.
C          (I*4)  J       = LOOP INDEX.
C
```

```
C          (R*8)  Z1      = REAL VALUE = IZ1.
C          (R*8)  TI      = TEMPERATURE (ION DISTRIBUTION).
C                             UNITS:
C          (R*8)  XNI     = REAL VALUE = NI.
C          (R*8)  XLI     = REAL VALUE = LI.
C          (R*8)  XLJ     = REAL VALUE = LJ.
C          (R*8)  XLG     =
C          (R*8)  FACT1   =
C          (R*8)  FACT2   =
C          (R*8)  EI0     = BINDING ENERGY FOR STATE I.
C                             UNITS: RYD
C          (R*8)  EJ0     = BINDING ENERGY FOR STATE J.
C                             UNITS: RYD
C          (R*8)  ZEFFI   = EFFECTIVE ION CHARGE FOR STATE I.
C          (R*8)  ZEFFJ   = EFFECTIVE ION CHARGE FOR STATE J.
C          (R*8)  TAUI    = RADIATIVE LIFETIME FOR STATE I.
C                             UNITS: SECS
C          (R*8)  TAUJ    = RADIATIVE LIFETIME FOR STATE J.
C                             UNITS: SECS
C          (R*8)  TAU     =
C          (R*8)  XS      =
C          (R*8)  XXJI    =
C          (R*8)  XXJJ    =
C          (R*8)  DE      =
C          (R*8)  W       =
C          (R*8)  T1      =
C          (R*8)  T2      =
C          (R*8)  T3      =
C          (R*8)  F1      =
C          (R*8)  F2      =
C
C          (R*8)  XSI()   =
C                             DIMENSION: 2
C          (R*8)  XJI()   =
C                             DIMENSION: 2
C          (R*8)  XEI()   =
C                             DIMENSION: 2
C          (R*8)  XSJ()   =
C                             DIMENSION: 2
C          (R*8)  XJJ()   =
C                             DIMENSION: 2
C          (R*8)  XEJ()   =
C                             DIMENSION: 2
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4UNIT     ADAS      RETURNS UNIT NO. FOR OUTPUT OF MESSAGES.
C          I4IDFL     ADAS      RETURNS UNIQUE INDEX GIVEN QUANTUM
C                               NUMBERS N AND L.
C          R8ZETA     ADAS
C          R8WIG6     ADAS
C          CXHYDE     ADAS      CALCULATES BINDING ENERGY FOR H-LIKE
C                               ION.
C          CXLTHE     ADAS      CALCULATES BINDING ENERGY FOR LI-LIKE
C                               ION.
C          CXSODE     ADAS      CALCULATES BINDING ENERGY FOR NA-LIKE
C                               ION.
C
C NOTES:
C       1) THE J->J' TRANSITION INDEX IS AS FOLLOWS:
```

```
C              1 : J=L+0.5 -> J'=L'+0.5
C              2 : J=L+0.5 -> J'=L'-0.5
C              3 : J=L-0.5 -> J'=L'+0.5
C              4 : J=L-0.5 -> J'=L'-0.5
C
C AUTHOR:  JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 5183
C
C DATE:    04/11/93
C
C UNIX-IDL PORT:
C
C AUTHOR:  WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C
C DATE:    22ND MAY 1996
C
C VERSION: 1.1                          DATE: 22-05-96
C MODIFIED: WILLIAM OSBORN
C               - FIRST VERSION. IBM VERSION NOT CHANGED
C
C VERSION: 1.2                          DATE: 17-05-07
C MODIFIED: Allan Whiteford
C               - Updated comments as part of subroutine documentation
C                 procedure.
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IZ0,          IZ1,          LI,           LJ
      INTEGER             MXJSHL,       MXNSHL,       NI,           NJ
      REAL*8              AMSSNO,       BMAG,         FMI(2*MXJSHL)
      REAL*8              FMJ(2*MXJSHL),              FMM(2*MXJSHL)
      REAL*8              FMP(2*MXJSHL)
      REAL*8              TBLF((MXNSHL*(MXNSHL+1))/2),            TIEV
```

4.90 cxphot: Subroutine cxphot from library adas3xx

```
      SUBROUTINE CXPHOT( IZ1    , TE     , TR     , V      ,
     &                   N      , L      , L1     , LP     ,
     &                   ISP    , LT     , LT1    , IS     ,
     &                   IRES   , PREC   , PION   , PSTIM
     &                 )
C
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: CXPHOT *********************
C
C  PURPOSE:  CALCULATES PHOTO INTEGRALS USING GIIH BOUND-FREE
C            GAUNT-FACTORS.
C
C  CALLING PROGRAM: C6TBRC.
C
C  INPUT : (I*4)  IZ1     = ION CHARGE.
C  INPUT : (R*8)  TE      = ELECTRON TEMPERATURE.
C                           UNITS: K
C  INPUT : (R*8)  TR      = RADIATION TEMPERATURE.
C                           UNITS: K
C  INPUT : (R*8)  V       = EFFECTIVE PRINCIPAL QUANTUM NUMBER OF BOUND
C                           ELECTRON.
C  INPUT : (I*4)  N       = PRINCIPAL QUANTUM NUMBER OF BOUND ELECTRON.
C  INPUT : (I*4)  L       = ORBITAL QUANTUM NUMBER OF BOUND ELECTRON.
C  INPUT : (I*4)  L1      = ORBITAL QUANTUM NUMBER OF FREE ELECTRON.
C  INPUT : (I*4)  LP      = TOTAL ORBITAL ANGULAR MOMENTUM QUANTUM
C                           NUMBER OF PARENT STATE.
C  INPUT : (I*4)  ISP     = 2*SP+1 WHERE SP IS TOTAL SPIN OF PARENT
C                           STATE.
C  INPUT : (I*4)  LT      = TOTAL ORBITAL ANGULAR MOMENTUM QUANTUM
C                           NUMBER OF BOUND SYSTEM.
C  INPUT : (I*4)  LT1     = TOTAL ORBITAL ANGULAR MOMENTUM QUANTUM
C                           NUMBER OF FREE SYSTEM.
C  INPUT : (I*4)  IS      = 2*S+1 WHERE S IS TOTAL SPIN OF SYSTEM.
C  INPUT : (I*4)  IRES    = LEVEL OF RESOLUTION.
C                         = 1 :
C                         = 2 : ABOVE LT1 SUM.
C                         = 3 : ABOVE LT SUM.
C                         = 4 : ABOVE S SUM.
C                         = 5 : UNRESOLVED GBF.
C
C  OUTPUT: (R*8)  PREC    = RADIATIVE RECOMBINATION INTEGRAL.
C  OUTPUT: (R*8)  PION    = PHOTOIONISATION INTEGRAL.
C  OUTPUT: (R*8)  PSTIM   = STIMULATED RECOMBINATION INTEGRAL.
C
C
C  PARAM : (I*4)  MXT     = 8.
C  PARAM : (I*4)  MXI     = 4.
C  PARAM : (R*8)  P1      =
C  PARAM : (R*8)  P2      = 0.9
C
C          (I*4)  IB      = FLAGS VALUE OF B.
C                         = 1 : B >= P2.
C                         = 2 : B <  P2.
C          (I*4)  IB1     = FLAGS VALUE OF B1.
C                         = 1 : B1 >= P2.
C                         = 2 : B1 <  P2.
C          (I*4)  I       = LOOP INDEX.
```

```
C
C          (R*8)  Z1      = REAL VALUE = IZ1.
C          (R*8)  B       = P1 * Z1**2 / ( V**2 * TE )
C          (R*8)  B1      = P1 * Z1**2 / ( V**2 * TR )
C          (R*8)  VVE     =
C          (R*8)  GI      = R8GIIH().
C          (R*8)  GI1     = R8GIIH().
C          (R*8)  GI2     = R8GIIH().
C          (R*8)  F1      =
C          (R*8)  F2      =
C          (R*8)  F3      =
C          (R*8)  F4      =
C          (R*8)  F5      =
C          (R*8)  F6      =
C          (R*8)  T       =
C          (R*8)  T2      =
C          (R*8)  T4      =
C          (R*8)  T6      =
C          (R*8)  Q1      =
C          (R*8)  Q2      =
C          (R*8)  EX      =
C          (R*8)  EX1     =
C          (R*8)  D1      =
C          (R*8)  D2      =
C          (R*8)  U2      =
C          (R*8)  U6      =
C
C          (R*8)  X1()    =
C                           DIMENSION: MXT.
C          (R*8)  X2()    =
C                           DIMENSION: MXT.
C          (R*8)  W1()    =
C                           DIMENSION: MXT.
C          (R*8)  W2()    =
C                           DIMENSION: MXT.
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          R8GIIH     ADAS      RETURNS BOUND-FREE G-FACTORS.
C          R8FEII     ADAS
C
C AUTHOR:  JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 5183
C
C DATE:    05/11/93
C
C UNIX-IDL PORT:
C
C AUTHOR:  WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C
C DATE:    22ND MAY 1996
C
C VERSION: 1.1                          DATE: 22-05-96
C MODIFIED: WILLIAM OSBORN
C               - FIRST VERSION. IBM VERSION NOT CHANGED
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
```

```
      INTEGER            IRES,       IS,         ISP,        IZ1
      INTEGER            L,          L1,         LP,         LT
      INTEGER            LT1,        N
      REAL*8             PION,       PREC,       PSTIM,      TE
      REAL*8             TR,         V
```

4.91 cxpint: Subroutine cxpint from library adas3xx

```
      SUBROUTINE CXPINT( MXTERM ,
     &                   C      , NC     , IPC    , UC     ,
     &                   D      , ND     , IPD    , UD
     &                 )
C
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: CXPINT *********************
C
C  PURPOSE: INTEGRATES POWER SERIES OF THE FORM:
C
C           X**IPC * EXP( UC * X ) * ( C(1) + X * C(2) + .... )
C                                    <------ NC TERMS ------>
C
C           TO GIVE AN ANSWER OF THE FORM:
C
C           -X**IPD * EXP( UD * X ) * ( D(1) + X * D(2) + .... )
C                                     <------ ND TERMS ------>
C
C           NOTE SIGN OF OUTPUT.
C
C  CALLING PROGRAM: ADAS308.
C
C  INPUT : (I*4)  MXTERM  = MAXIMUM NUMBER OF TERMS.
C  INPUT : (I*4)  NC      = NUMBER OF TERMS IN POWER SERIES.
C  INPUT : (I*4)  IPC     = POWER OF LEADING TERM.
C  INPUT : (R*8)  UC      = POWER OF EXPONENT.
C  INPUT : (R*8)  C()     = POWER SERIES COEFFICIENTS.
C
C  OUTPUT: (I*4)  ND      = NUMBER OF TERMS IN POWER SERIES.
C  OUTPUT: (I*4)  IPD     = POWER OF LEADING TERM.
C  OUTPUT: (R*8)  UD      = POWER OF EXPONENT.
C  OUTPUT: (R*8)  D()     = POWER SERIES COEFFICIENTS.
C
C  PARAM : (R*8)  P1      = 1.0D+12
C
C          (I*4)  I       = LOOP INDEX.
C
C          (R*8)  T       =
C          (R*8)  T1      =
C
C ROUTINES: NONE
C
C AUTHOR:   JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C           K1/0/81
C           JET EXT. 5183
C
C DATE:     01/10/93
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IPC,         IPD,         MXTERM,      NC
      INTEGER             ND
      REAL*8              C(MXTERM),   D(MXTERM),   UC,          UD
```

4.92 cxpmat: Subroutine cxpmat from library adas3xx

```
SUBROUTINE  CXPMAT(PRMAT,NRES,MAXTE,MAXNE,MAXEB,EBRAY,TERAY,
     &    NERAY,NB,RCMAT,TEREF, NEREF,EBREF,ITREF,
     &    INREF,IEREF,IUNIT,INFILE,INA,ITA,IEA,
     &    NBENG,NTEMP,NDENS,NLEV)
C-----------------------------------------------------------------------
C
C ********** FORTRAN 77 ROUTINE : CXPMAT.FOR **********
C
C   PURPOSE : TO INTERROGATE AND EXTRACT THE  PROJECTION MATRICES
C       WHICH ARE GENERATED USING ADAS311
C
C   INPUT   :
C
C
C     (I*4)       INUNIT : STREAM NUMBER FOR ACCESSING
C   INPUT FILE.
C     (I*4) NBENG : MAXIMUM NUMBER OF ENERGIES.
C     (I*4) NDENS : MAXIMUM NUMBER OF DENSITIES.
C     (I*4) NTEMP : MAXIMUM NUMBER OF TEMPS.
C     (I*4) NLEV : MAXIMUM NUMBER OF LEVELS
C   CONTAINED IN THE PROJECTION
C   AND RECOMBINATION MATRICES.
C     (CHR) INFILE : INPUT FILENAME TO BE
C   INTERROGATED.
C
C   OUTPUT  :
C
C     (R*8) PRMAT() : PROJECTION MATRIX.
C   1ST DIM.: LEVEL INDEX.
C   2ND DIM.: LEVEL INDEX.
C   3RD DIM.: ENERGY INDEX.
C   4TH DIM.: DENSITY INDEX.
C   5TH DIM.: TEMPERATURE INDEX.
C     (R*8) RCMAT() : COLLISIONAL-RADIATIVE
C   RECOMBINATION COEFFICIENTS.
C   1ST DIM.: LEVEL INDEX.
C   2ND DIM.: ENERGY INDEX.
C   3RD DIM.: DENSITY INDEX.
C   4TH DIM.: TEMPERATURE INDEX.
C     (R*8) EBRAY() : ARRAY CONTAINING THE NEUTRAL
C           BEAM ENERGY (eV/amu).
C   1ST DIM.: ENERGY INDEX.
C     (R*8) TERAY() : ARRAY CONTAINING THE
C   TEMPERATURE (eV).
C   1ST DIM.: TEMPERATURE INDEX.
C     (R*8) NERAY() : ARRAY CONTAINING THE ELECTRON
C   DENSITY.
C   1ST DIM.: DENSITY INDEX.
C     (R*8) TEREF : REFERENCE TEMPERATURE (eV).
C     (R*8) NEREF : REFERENCE ELECTRON DENSITY (cm-3).
C     (R*8) EBREF : REFERENCE BEAM ENERGY (eV amu-1).
C     (R*8) NB : NEUTRAL BEAM DENSITY (cm-3)
C
C     (I*4) NRES : NUMBER OF RESOLVED LEVELS
C   CONTAINED IN THE COLLISIONAL
C   RADIATIVE MATRIX.
C     (I*4) MAXTE : NUMBER OF TEMPERATURES CONTAINED
C   IN THE INPUT FILE.
C     (I*4) MAXNE : NUMBER OF DENSITIES CONTAINED
```

```
C   IN THE INPUT FILE.
C     (I*4) MAXEB : NUMBER OF BEAM ENERGIES CONTAINED
C   IN THE INPUT FILE.
C     (I*4) ITREF : INDEX FOR REFERENCE TEMPERATURE.
C   TO BE USED INCONJUCTION WITH THE
C   ARRAY ITA().
C     (I*4) INREF : INDEX FOR REFERENCE DENSITY.
C   TO BE USED INCONJUCTION WITH THE
C   ARRAY INA().
C     (I*4) IEREF : INDEX FOR REFERENCE BEAM ENERGY.
C   TO BE USED INCONJUCTION WITH THE
C   ARRAY IEA().
C     (I*4) INA() : ARRAY CONTAINING THE DENSITY
C   INDEX.
C     (I*4) IEA() : ARRAY CONTAINING THE ENERGY
C   INDEX.
C     (I*4) ITA() : ARRAY CONTAINING THE
C   TEMPERATURE INDEX.
C
C  GENERAL  :
C
C     (R*8) EB : NEUTRAL BEAM ENERGY (eV/AMU).
C     (R*8) TE : ELECTRON TEMPERATURE (eV).
C     (R*8) TP : ION TEMPERATURE (K).
C     (R*8) NE : ELECTRON DENSITY (cm-3).
C     (R*8) NP : ION DENSITY (cm-3).
C
C     (I*4) NTOTAL : COUNTER USED TO DETERMINE HOW
C   MANY TABULATED TABLES TO BE
C   READ FROM INPUT FILE.
C     (I*4)  I : GENERAL COUNTER.
C     (I*4)       J : GENERAL COUNTER.
C     (I*4) K : GENERAL COUNTER.
C     (I*4) IN : GENERAL INDEX.
C     (I*4) IT : GENERAL INDEX.
C     (I*4) IE : GENERAL INDEX.
C     (I*4) ITCOUNT : NUMBER OF TEMPERATURES.
C     (I*4) INCOUNT : NUMBER OF DENSITIES.
C     (I*4) IECOUNT         : NUMBER OF BEAM ENERGIES.
C     (I*4) ICOUNT : GENERAL COUNTER.
C     (CHR) LINE : GENERAL VARIABLE.
C
C
C
C
C   ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          CCFIND     ADAS    IDENTIFIED REPEATED VALUES IN ARRAY.
C    CCSORT     ADAS    SORTS ARRAY VALUES INTO INCREASING ORDER.
C    CCFILL     ADAS    FILLS ARRAY WITH CORRESPONDING INDEX.
C
C
C
C   AUTHOR  : HARVEY ANDERSON
C       UNIVERSITY OF STRATHCLYDE
C       ANDERSON@PHYS.STRATH.AC.UK
C
C   DATE    : 30/8/99
C
C  VERSION  : 1.2
```

```
C  DATE     : 23-01-2001
C
C  MODIFIED : Martin O'Mullane
C             Declaration of variables is not standard. NTEMP is
C             defined after it is first used to dimension a
C             variable length input array. Reconfigure variable
C             definitions to follow ADAS conventions.
C
C-----------------------------------------------------------------------
      CHARACTER*80        INFILE
      INTEGER             IEA(NBENG),  IEREF,       INA(NDENS),  INREF
      INTEGER             ITA(NTEMP),  ITREF,       IUNIT,       MAXEB
      INTEGER             MAXNE,       MAXTE,       NBENG,       NDENS
      INTEGER             NLEV,        NRES,        NTEMP
      REAL*8              EBRAY(NBENG),             EBREF,       NB
      REAL*8              NERAY(NDENS),             NEREF
      REAL*8              PRMAT(NLEV,NLEV,NBENG,NDENS,NTEMP)
      REAL*8              RCMAT(NLEV,NBENG,NDENS,NTEMP)
      REAL*8              TERAY(NTEMP),             TEREF
```

4.93 cxpprd: Subroutine cxpprd from library adas3xx

```
      SUBROUTINE CXPPRD( MXTERM ,
     &                   A      , NA     , IPA    ,
     &                   B      , NB     , IPB    ,
     &                   C      , NC     , IPC
     &                 )
C
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: CXPPRD *********************
C
C  PURPOSE: CALCULATES PRODUCT OF TWO POWER SERIES OF THE FORM:
C
C           X**IPA * ( A(1) + X * A(2) + .... )
C                    <------ NA TERMS ------>
C
C  CALLING PROGRAM: ADAS308.
C
C  INPUT : (I*4)  MXTERM  = MAXIMUM NUMBER OF TERMS.
C  INPUT : (I*4)  NA      = NUMBER OF TERMS IN FIRST POWER SERIES.
C  INPUT : (I*4)  IPA     = POWER OF LEADING TERM IN FIRST SERIES.
C  INPUT : (R*8)  A()     = POWER SERIES COEFFICIENTS IN FIRST SERIES.
C  INPUT : (I*4)  NB      = NUMBER OF TERMS IN SECOND POWER SERIES.
C  INPUT : (I*4)  IPB     = POWER OF LEADING TERM IN SECOND SERIES.
C  INPUT : (R*8)  B()     = POWER SERIES COEFFICIENTS IN SECOND SERIES.
C
C  OUTPUT: (I*4)  NC      = NUMBER OF TERMS IN POWER SERIES.
C  OUTPUT: (I*4)  IPC     = POWER OF LEADING TERM.
C  OUTPUT: (R*8)  C()     = POWER SERIES COEFFICIENTS.
C
C          (I*4)  I       = LOOP INDEX.
C          (I*4)  J       = LOOP INDEX.
C          (I*4)  JU      = LOOP LIMIT.
C          (I*4)  JL      = LOOP LIMIT.
C
C          (R*8)  X       = STORE FOR SUM WHEN CALCULATING C(I).
C
C ROUTINES: NONE
C
C AUTHOR:   JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C           K1/0/81
C           JET EXT. 5183
C
C DATE:     01/10/93
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IPA,          IPB,          IPC,          MXTERM
      INTEGER             NA,           NB,           NC
      REAL*8              A(MXTERM),    B(MXTERM),    C(MXTERM)
```

4.94 cxqcx: Subroutine cxqcx from library adas3xx

```
      SUBROUTINE CXQCX(DSNIN  , NEIN  , EIN   ,
     &                 INSEL  , ILSEL ,
     &                 CXOUT  , LEXT  )

C-----------------------------------------------------------------------
C
C  ******************** FORTRAN 77 SUBROUTINE: CXQCX *****************
C
C  PURPOSE: Gathers data from adf01 charge exchange files and
C           interpolates on the requested energy vector.
C
C
C  INPUT
C
C  (C*80) DSNIN    : adf01 file name.
C  (R*8)  NEIN     : NUMBER OF USER REQUESTED ENERGIES
C                    UNITS: EV/AMU
C  (R*8)  EIN      : USER REQUESTED ENERGIES
C  (I*4)  INSEL    : SELECTED INPUT DATA n QUANTUM SHELL - 0 for total
C  (I*4)  ILSEL    : SELECTED INPUT DATA l QUANTUM SHELL
C
C
C
C  OUTPUT
C
C  (R*8)  CXOUT()  : CROSS SECTION DATA
C                    UNITS: CM**2
C  (L*4)  LEXT()   : .TRUE. IF INTEPOLATED
C
C
C  PROGRAM:
C
C
C  ROUTINES:
C
C     ROUTINE    SOURCE    BRIEF DESCRIPTION
C     ------------------------------------------------------------
C     CXDATA     ADAS      READS INPUT DATA SET IN ADF01 FORMAT.
C     C1BSIG     ADAS      SETS UP X-SECTIONS FOR SELECTED N-SHELL
C     XXSPLE     ADAS      SPLINE SUBROUTINE (EXTENDED DIAGNOSTICS)
C     R8FUN1     ADAS      REAL*8 FUNCTION: ( X -> X )
C
C
C  NOTE     : Depending on the choice of output cross section the
C             appropriate n,l or m value must be present.
C
C
C  VERSION  : 1.1
C  DATE     : 18-01-2001
C  AUTHOR   : Martin O'Mullane
C
C  DATE     : 14-03-2007
C  VERSION  : 1.2
C  MODIFIED : Martin O'Mullane
C               - Increase maximum number of shells to 100.
C               - Use xxdata_01 to read in adf01 dataset.
C
C  VERSION  : 1.3
C  DATE     : 22-05-2007
```

```
C  MODIFIED : Martin O'Mullane
C               - Remove unused m-subshell data possibility.
C
C-----------------------------------------------------------------------
      CHARACTER*80        DSNIN
      INTEGER             ILSEL,       INSEL,       NEIN
      LOGICAL             LEXT(NEIN)
      REAL*8              CXOUT(NEIN), EIN(NEIN)
```

4.95 cxqeik: Subroutine cxqeik from library adas3xx

```
      SUBROUTINE CXQEIK( MXNSHL , MXBEAM , IZ1    , IDONOR ,
     &                   NBOT   , NTOP   , NBEAM  , BMENA  ,
     &                   BMFRA  , QTHEOR , FTHEOR
     &                 )
C
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: CXQEIK *********************
C
C  PURPOSE:  USES THE EIKONIAL APPROXIMATION TO CALCULATE THE
C            THEORETICAL CHARGE EXCHANGE RATE COEFFICIENTS TO N SHELLS
C            AND THE NL FRACTIONS FROM NEUTRAL HYDROGEN OR HELIUM IN
C            GROUND OR EXCITED STATE TO A BARE NUCLEUS TARGET.
C
C            AN ENERGY DEPENDENT MODIFYING FACTOR CAN BE SWITCHED ON TO
C            MAKE THE TOTAL RATE COEFFT. AGREE BETTER WITH UDWA AT LOW
C            ENERGY. THIS IS ESTABLISHED FROM H+C(+6) AND H+O(+8) DATA.
C            LMOD=.TRUE. SWITCHES ON THE MODIFICATION.
C
C  CALLING PROGRAM: ADAS308 , C6QEIK.
C
C  INPUT : (I*4)  MXNSHL   = MAXIMUM VALUE OF N QUANTUM NUMBER.
C  INPUT : (I*4)  MXBEAM   = MAXIMUM NUMBER OF BEAM ENERGIES.
C  INPUT : (R*8)  IZ1      = CHARGE OF TARGET ION.
C  INPUT : (I*4)  IDONOR   = DONOR STATE FOR EIKONAL MODEL.
C                            1 = H(1S)    DONOR
C                            2 = H(2S)    DONOR
C                            3 = H(2P)    DONOR
C                            4 = H(1S2)   DONOR
C                            5 = HE(1S2S) DONOR
C  INPUT : (I*4)  NBOT     = MINIMUM PRINCIPAL QUANTUM NUMBER.
C  INPUT : (I*4)  NTOP     = MAXIMUM PRINCIPAL QUANTUM NUMBER.
C  INPUT : (I*4)  NBEAM    = NO. OF ENERGY COMPONENTS IN NEUTRAL BEAM.
C  INPUT : (R*8)  BMENA()  = BEAM ENERGY COMPONENTS.
C                            UNITS: EV/AMU
C                            DIMENSION: COMPONENT INDEX.
C  INPUT : (R*8)  BMFRA()  = BEAM COMPONENT FRACTIONS.
C                            DIMENSION: COMPONENT INDEX.
C
C  OUTPUT: (R*8)  QTHEOR() = MEAN RATE COEFFICIENTS FOR N-LEVELS
C                            AVERAGED OVER BEAM FRACTIONS.
C                            UNITS: CM3 SEC-1
C                            DIMENSION: REFERENCED BY N QUANTUM NUMBER.
C  OUTPUT: (R*8)  FTHEOR() = MEAN RATE FOR NL-LEVELS AS A FRACTION OF
C                            CORRESPONDING N-LEVEL.
C                            DIMENSION: REFERENCED BY FUNC I4IDFL(N,L).
C
C  PARAM : (I*4)  MXN      = 'MXNSHL'.
C  PARAM : (R*8)  P1       =
C  PARAM : (R*8)  P2       =
C  PARAM : (R*8)  P3       =
C  INPUT : (L*4)  LMOD     = MODIFY FLAG.
C                            .TRUE.  = MODIFY RATE COEFFICIENTS.
C                            .FALSE. = LEAVE COEFFICIENTS UNCHANGED.
C
C          (I*4)  NA       = PRINCIPAL QUANTUM NUMBER OF ELECTRON IN
C                            INITIAL STATE OF INCIDENT NEUTRAL ATOM.
C          (I*4)  LA       = ORBITAL QUANTUM NUMBER OF ELECTRON IN
```

```
C                             INITIAL STATE OF INCIDENT NEUTRAL ATOM.
C          (I*4)  N         = PRINCIPAL QUANTUM NUMBER OF FINAL STATE.
C          (I*4)  L         = ORBITAL QUANTUM NUMBER.
C          (I*4)  IB        = ENERGY INDEX.
C          (I*4)  IDL       = INDEX FROM FUNC I4IDFL(N,L).
C
C          (R*8)  ZT        = SCREENING CHARGE FOR THE 1S ELECTRON OF THE
C                             TARGET ATOM IN THE INITIAL STATE.
C          (R*8)  ZT1       = EFFECTIVE CHARGE FOR THE 1S ELECTRON OF THE
C                             TARGET ATOM IN THE FINAL STATE.
C          (R*8)  THETA     = PARAMETER TO GIVE CORRECT BINDING ENERGY
C                             FOR INITIAL TARGET STATE.
C          (R*8)  VEL       = VELOCITY OF INCIDENT ATOM.
C                             UNITS: CM SEC-1
C          (R*8)  VELAU     = VELOCITY OF INCIDENT ATOM.
C                             UNITS: AT. UNITS.
C          (R*8)  XSECNA    = N-RESOLVED CROSS-SECTION FOR CAPTURE.
C                             UNITS: AT. UNITS
C          (R*8)  DIV       = DIVISOR FOR CROSS-SECTIONS.
C
C          (R*8)  FRACLA()  = L-RESOLVED CROSS-SECTION AS A FRACTION OF
C                             CORRESPONDING N-RESOLVED CROSS-SECTION.
C                             DIMENSION: REFERENCED BY L QUANTUM NUMBER.
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4UNIT     ADAS      RETURNS UNIT NO. FOR OUTPUT OF MESSAGES.
C          I4IDFL     ADAS      RETURNS UNIQUE INDEX GIVEN QUANTUM
C                               NUMBERS N AND L.
C          CXSGEI     ADAS      CALCULATES EXCHANGE RATE CROSS-SECTIONS.
C
C AUTHOR:   JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C           K1/0/81
C           JET EXT. 5183
C
C DATE:     07/10/93
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 20-06-95
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - PUT UNDER S.C.C.S. CONTROL
C
C VERSION: 1.2                          DATE: 02-04-96
C MODIFIED: TIM HAMMOND
C               - COMMENTED OUT UNREACHABLE LINES WHEN LMOD IS SET
C                 TO .FALSE. WHICH IS CURRENTLY THE CASE (THIS PREVENTS
C                 THE COMPILER GIVING INFO MESSAGES).
C
C VERSION: 1.3                          DATE: 17-05-07
C MODIFIED: Allan Whiteford
C               - Corrected typo in comments.
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IDONOR,      IZ1,         MXBEAM,      MXNSHL
      INTEGER             NBEAM,       NBOT,        NTOP
      REAL*8              BMENA(MXBEAM),            BMFRA(MXBEAM)
      REAL*8              FTHEOR((MXNSHL*(MXNSHL+1))/2)
```

```
REAL*8               QTHEOR(MXNSHL)
```

4.96 cxqxch: Subroutine cxqxch from library adas3xx

```
      SUBROUTINE CXQXCH ( MXNENG , MXNSHL , MXBEAM , NBEAM  ,
     &                    BMENA  , BMFRA  , NBOT   , NTOP   ,
     &                    NMINF  , NMAXF  , NENRGY , ENRGYA ,
     &                    ALPHAA , XSECNA , FRACLA , QTHEOR ,
     &                    FTHEOR
     &                  )
C
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: CXQXCH *********************
C
C  PURPOSE:  USES THE INPUT DATASET TO CALCULATE THE CHARGE EXCHANGE
C            RATE COEFFICIENTS FOR BOTH N-LEVELS AND NL-LEVELS AVERAGED
C            OVER THE BEAM FRACTIONS.
C
C            NL-LEVEL RATES ARE EXPRESSED AS A FRACTION OF
C            CORRESPONDING N-LEVEL.
C
C  CALLING PROGRAM: ADAS308 , C6QXCH
C
C  INPUT : (I*4)  MXNENG    = MAXIMUM NO. OF ENERGIES.
C  INPUT : (I*4)  MXNSHL    = MAXIMUM NO. OF N SHELLS.
C  INPUT : (I*4)  MXBEAM    = MAXIMUM NO. OF BEAM ENERGIES.
C  INPUT : (I*4)  NBEAM     = NUMBER OF BEAM ENERGIES.
C  INPUT : (R*8)  BMENA()   = BEAM ENERGY COMPONENTS.
C                             UNITS: EV/AMU
C                             DIMENSION: COMPONENT INDEX.
C  INPUT : (R*8)  BMFRA()   = BEAM COMPONENT FRACTIONS.
C                             DIMENSION: COMPONENT INDEX.
C  INPUT : (I*4)  NBOT      = MINIMUM PRINCIPAL QUANTUM NUMBER.
C  INPUT : (I*4)  NTOP      = MAXIMUM PRINCIPAL QUANTUM NUMBER.
C  INPUT : (I*4)  NMINF     = MINIMUM PRINCIPAL QUANTUM NUMBER OF INPUT
C                             DATASET.
C  INPUT : (I*4)  NMAXF     = MAXIMUM PRINCIPAL QUANTUM NUMBER OF INPUT
C                             DATASET.
C  INPUT : (I*4)  NENRGY    = NUMBER OF ENERGIES IN DATASET.
C  INPUT : (R*8)  ENRGYA()  = COLLISION ENERGIES.
C                             UNITS: EV/AMU
C                             DIMENSION: ENERGY INDEX
C  INPUT : (R*8)  ALPHAA()  = EXTRAPOLATION PARAMETER ALPHA.
C                             DIMENSION: ENERGY INDEX
C  INPUT : (R*8)  XSECNA(,) = N-RESOLVED CHARGE EXCHANGE CROSS-SECTIONS.
C                             UNITS: CM2
C                             1ST DIMENSION: ENERGY INDEX
C                             2ND DIMENSION: N-SHELL
C  INPUT : (R*8)  FRACLA(,) = L-RESOLVED CHARGE EXCHANGE CROSS-SECTIONS
C                             EXPRESSED AS FRACTION OF CORRESPONDING
C                             N-RESOLVED CROSS-SECTION.
C                             1ST DIMENSION: ENERGY INDEX
C                             2ND DIMENSION: REFERENCED BY I4IDFL(N,L)
C
C  OUTPUT: (R*8)  QTHEOR()  = MEAN RATE COEFFICIENTS FOR N-LEVELS
C                             AVERAGED OVER BEAM FRACTIONS.
C                             UNITS: CM3 SEC-1
C                             DIMENSION: REFERENCED BY N QUANTUM NUMBER.
C  OUTPUT: (R*8)  FTHEOR()  = MEAN RATE COEFFICIENTS FOR NL-LEVELS
C                             AVERAGED OVER BEAM FRACTIONS. EXPRESSED AS
C                             FRACTIONS OF CORRESPONDING N-LEVELS.
```

```
C                             DIMENSION: REFERENCED BY I4IDFL(N,L)
C
C  PARAM : (I*4)  MXB       = 'MXBEAM'.
C  PARAM : (I*4)  MXN       = 'MXNSHL'.
C
C          (I*4)  IB        = BEAM INDEX.
C
C          (R*8)  RATE(,)   = RATE COEFFICIENTS FOR EACH COMPONENT OF
C                             THE BEAM AS A FUNCTION OF N-LEVEL.
C                             UNITS: CM3 SEC-1
C                             1ST DIMENSION: BEAM INDEX
C                             2ND DIMENSION: N-SHELL
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4UNIT     ADAS      RETURN UNIT NO. FOR OUTPUT OF MESSAGES.
C          CXQXN      ADAS      CALCULATES N-LEVEL RATE COEFFICIENTS.
C          CXQXL      ADAS      CALCULATES NL-LEVEL RATE COEFFICIENTS AS
C                               FRACTION OF CORRESPONDING N-LEVEL.
C
C AUTHOR:  JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 5183
C
C DATE:    19/10/93
C
C
C VERSION: 1.1                          DATE: 20-06-95
C MODIFIED: TIM HAMMOND (Probably)
C               - PUT UNDER S.C.C.S. CONTROL
C
C VERSION  : 1.2                        DATE: 17-05-07
C MODIFIED : Allan Whiteford
C               - Updated comments as part of subroutine documentation
C                 procedure.
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             MXBEAM,      MXNENG,      MXNSHL,      NBEAM
      INTEGER             NBOT,        NENRGY,      NMAXF,       NMINF
      INTEGER             NTOP
      REAL*8              ALPHAA(MXNENG),           BMENA(MXBEAM)
      REAL*8              BMFRA(MXBEAM),            ENRGYA(MXNENG)
      REAL*8              FRACLA(MXNENG,(MXNSHL*(MXNSHL+1))/2)
      REAL*8              FTHEOR((MXNSHL*(MXNSHL+1))/2)
      REAL*8              QTHEOR(MXNSHL)
      REAL*8              XSECNA(MXNENG,MXNSHL)
```

4.97 cxqxl: Subroutine cxqxl from library adas3xx

```
      SUBROUTINE CXQXL ( MXNENG , MXNSHL , MXBEAM , NBEAM  ,
     &                   BMENA  , BMFRA  , NBOT   , NTOP   ,
     &                   NENRGY , ENRGYA , FRACLA , QTHEOR ,
     &                   RATE   , FTHEOR
     &                 )
C
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: CXQXL  *********************
C
C  PURPOSE:  USES THE INPUT DATASET TO CALCULATE THE CHARGE EXCHANGE
C            RATE COEFFICIENTS FOR NL-LEVELS AVERAGED OVER THE BEAM
C            FRACTIONS. RATES ARE EXPRESSED AS A FRACTION OF
C            CORRESPONDING N-LEVEL.
C
C  CALLING PROGRAM: ADAS308
C
C  INPUT : (I*4)  MXNENG    = MAXIMUM NO. OF ENERGIES.
C  INPUT : (I*4)  MXNSHL    = MAXIMUM NO. OF N SHELLS.
C  INPUT : (I*4)  MXBEAM    = MAXIMUM NO. OF BEAM ENERGIES.
C  INPUT : (I*4)  NBEAM     = NUMBER OF BEAM ENERGIES.
C  INPUT : (R*8)  BMENA()   = BEAM ENERGY COMPONENTS.
C                             UNITS: EV/AMU
C                             DIMENSION: COMPONENT INDEX.
C  INPUT : (R*8)  BMFRA()   = BEAM COMPONENT FRACTIONS.
C                             DIMENSION: COMPONENT INDEX.
C  INPUT : (I*4)  NBOT      = MINIMUM PRINCIPAL QUANTUM NUMBER.
C  INPUT : (I*4)  NTOP      = MAXIMUM PRINCIPAL QUANTUM NUMBER.
C  INPUT : (I*4)  NENRGY    = NUMBER OF ENERGIES IN DATASET.
C  INPUT : (R*8)  ENRGYA()  = COLLISION ENERGIES.
C                             UNITS: EV/AMU
C                             DIMENSION: ENERGY INDEX
C  INPUT : (R*8)  FRACLA(,) = L-RESOLVED CHARGE EXCHANGE CROSS-SECTIONS
C                             EXPRESSED AS FRACTION OF CORRESPONDING
C                             N-RESOLVED CROSS-SECTION.
C                             1ST DIMENSION: ENERGY INDEX
C                             2ND DIMENSION: REFERENCED BY I4IDFL(N,L)
C  INPUT : (R*8)  QTHEOR()  = MEAN RATE COEFFICIENTS FOR N-LEVELS
C                             AVERAGED OVER BEAM FRACTIONS.
C                             UNITS: CM3 SEC-1
C                             DIMENSION: REFERENCED BY N QUANTUM NUMBER.
C  INPUT : (R*8)  RATE(,)   = RATE COEFFICIENTS FOR EACH COMPONENT OF
C                             THE BEAM AS A FUNCTION OF N-LEVEL.
C                             UNITS: CM3 SEC-1
C                             1ST DIMENSION: BEAM INDEX
C                             2ND DIMENSION: N-SHELL
C
C  INPUT : (R*8)  FTHEOR()  = MEAN RATE COEFFICIENTS FOR NL-LEVELS
C                             AVERAGED OVER BEAM FRACTIONS. EXPRESSED AS
C                             FRACTIONS OF CORRESPONDING N-LEVELS.
C                             DIMENSION: REFERENCED BY I4IDFL(N,L)
C
C  PARAM : (I*4)  MXN       = 'MXNSHL'.
C
C          (I*4)  IB        = BEAM INDEX.
C          (I*4)  N         = PRINCIPAL QUANTUM NUMBER.
C          (I*4)  IE        = ENERGY INDEX.
C          (I*4)  IEU       = ENERGY INDEX ABOVE BEAM ENERGY.
```

```
C          (I*4)  IEL       = ENERGY INDEX BELOW BEAM ENERGY.
C          (I*4)  IDL       = LOOP INDEX.
C
C          (R*8)  GRAD      = GRADIENT OF STRAIGHT LINE INTERPOLATION.
C
C          (L*4)  LMATCH    = FLAG IF BEAM ENERGY MATCHES EXACTLY A DATA
C                             SET ENERGY.
C                           = .TRUE.  => BEAM ENERGY MATCHES.
C                           = .FALSE. => BEAM ENERGY DOES NOT MATCH.
C
C          (R*8)  PTH()     = RATE COEFFICIENTS FOR NL-LEVELS FOR A
C                             PATICULAR BEAM COMPONENT.
C                             DIMENSION: REFERENCED BY I4IDFL(N,L)
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4UNIT     ADAS      RETURNS UNIT NO. FOR OUTPUT OF MESSAGES.
C          I4IDFL     ADAS      RETURNS UNIQUE INDEX GIVEN THE QUANTUM
C                               NUMBERS N AND L.
C
C AUTHOR:  JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 5183
C
C DATE:    19/10/93
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             MXBEAM,      MXNENG,      MXNSHL,      NBEAM
      INTEGER             NBOT,        NENRGY,      NTOP
      REAL*8              BMENA(MXBEAM),            BMFRA(MXBEAM)
      REAL*8              ENRGYA(MXNENG)
      REAL*8              FRACLA(MXNENG,(MXNSHL*(MXNSHL+1))/2)
      REAL*8              FTHEOR((MXNSHL*(MXNSHL+1))/2)
      REAL*8              QTHEOR(MXNSHL),           RATE(MXBEAM,MXNSHL)
```

4.98 cxqxn: Subroutine cxqxn from library adas3xx

```
      SUBROUTINE CXQXN ( MXNENG , MXNSHL , MXBEAM , NBEAM  ,
     &                   BMENA  , BMFRA  , NBOT   , NTOP   ,
     &                   NMINF  , NMAXF  , NENRGY , ENRGYA ,
     &                   ALPHAA , XSECNA , QTHEOR , RATE
     &                 )
C
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: CXQXN  *********************
C
C  PURPOSE:  USES THE INPUT DATASET TO CALCULATE THE CHARGE EXCHANGE
C            RATE COEFFICIENTS FOR N-LEVELS AVERAGED OVER THE BEAM
C            FRACTIONS.
C
C  CALLING PROGRAM: ADAS308
C
C  INPUT : (I*4)  MXNENG    = MAXIMUM NO. OF ENERGIEL.
C  INPUT : (I*4)  MXNSHL    = MAXIMUM NO. OF N SHELLS.
C  INPUT : (I*4)  MXBEAM    = MAXIMUM NO. OF BEAM ENERGIES.
C  INPUT : (I*4)  NBEAM     = NUMBER OF BEAM ENERGIES.
C  INPUT : (R*8)  BMENA()   = BEAM ENERGY COMPONENTS.
C                             UNITS: EV/AMU
C                             DIMENSION: COMPONENT INDEX.
C  INPUT : (R*8)  BMFRA()   = BEAM COMPONENT FRACTIONS.
C                             DIMENSION: COMPONENT INDEX.
C  INPUT : (I*4)  NBOT      = MINIMUM PRINCIPAL QUANTUM NUMBER.
C  INPUT : (I*4)  NTOP      = MAXIMUM PRINCIPAL QUANTUM NUMBER.
C  INPUT : (I*4)  NMINF     = MINIMUM PRINCIPAL QUANTUM NUMBER OF INPUT
C                             DATASET.
C  INPUT : (I*4)  NMAXF     = MAXIMUM PRINCIPAL QUANTUM NUMBER OF INPUT
C                             DATASET.
C  INPUT : (I*4)  NENRGY    = NUMBER OF ENERGIES IN DATASET.
C  INPUT : (R*8)  ENRGYA()  = COLLISION ENERGIES.
C                             UNITS: EV/AMU
C                             DIMENSION: ENERGY INDEX
C  INPUT : (R*8)  ALPHAA()  = EXTRAPOLATION PARAMETER ALPHA.
C                             DIMENSION: ENERGY INDEX
C  INPUT : (R*8)  XSECNA(,) = N-RESOLVED CHARGE EXCHANGE CROSS-SECTIONS.
C                             UNITS: CM2
C                             1ST DIMENSION: ENERGY INDEX
C                             2ND DIMENSION: N-SHELL
C
C  OUTPUT: (R*8)  QTHEOR()  = MEAN RATE COEFFICIENTS FOR N-LEVELS
C                             AVERAGED OVER BEAM FRACTIONS.
C                             UNITS: CM3 SEC-1
C                             DIMENSION: REFERENCED BY N QUANTUM NUMBER.
C  OUTPUT: (R*8)  RATE(,)   = RATE COEFFICIENTS FOR EACH COMPONENT OF
C                             THE BEAM AS A FUNCTION OF N-LEVEL.
C                             UNITS: CM3 SEC-1
C                             1ST DIMENSION: BEAM INDEX
C                             2ND DIMENSION: N-SHELL
C
C  PARAM : (R*8)  ALPMIN    = MINIMUM VALUE OF EXTRAPOLATION EXPONENT
C                             'ALPHA'.
C  PARAM : (R*8)  C1        =
C
C          (I*4)  IB        = BEAM INDEX.
C          (I*4)  N         = PRINCIPAL QUANTUM NUMBER.
```

```
C          (I*4)  IE        = ENERGY INDEX.
C          (I*4)  IEU       = ENERGY INDEX ABOVE BEAM ENERGY.
C          (I*4)  IEL       = ENERGY INDEX BELOW BEAM ENERGY.
C
C          (R*8)  XNMAXF    = REAL VALUE = NMAXF.
C          (R*8)  ALPHA     = EXTRAPOLATION EXPONENT.
C          (R*8)  VEL       = VELOCITY OF INCIDENT ATOM.
C                             UNITS: CM SEC-1
C          (R*8)  GRAD      = GRADIENT OF STRAIGHT LINE INTERPOLATION.
C          (R*8)  XSEC      = CROSS-SECTION AT INTERMEDIATE ENERGY.
C
C          (L*4)  LMATCH    = FLAG IF BEAM ENERGY MATCHES EXACTLY A DATA
C                             SET ENERGY.
C                           = .TRUE.  => BEAM ENERGY MATCHES.
C                           = .FALSE. => BEAM ENERGY DOES NOT MATCH.
C
C ROUTINES:  NONE
C
C AUTHOR:  JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 5183
C
C DATE:    18/10/93
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             MXBEAM,      MXNENG,      MXNSHL,      NBEAM
      INTEGER             NBOT,        NENRGY,      NMAXF,       NMINF
      INTEGER             NTOP
      REAL*8              ALPHAA(MXNENG),           BMENA(MXBEAM)
      REAL*8              BMFRA(MXBEAM),            ENRGYA(MXNENG)
      REAL*8              QTHEOR(MXNSHL),           RATE(MXBEAM,MXNSHL)
      REAL*8              XSECNA(MXNENG,MXNSHL)
```

4.99 cxsetp: Subroutine cxsetp from library adas3xx

```
      SUBROUTINE CXSETP( MXTAB  , MXGRF  , SYMBD  , IDZ0   ,
     &                   SYMBR  , IRZ0   , IRZ1   , IRZ2   ,
     &                   NGRND  , NTOT
     &                 )
C
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: CXSETP *********************
C
C  PURPOSE:  SETS UP PARAMETERS IN THE SHARED POOL FOR PANEL DISPLAY.
C
C  CALLING PROGRAM: ADAS306 / ADAS308.
C
C  DATA:  DATA IS OBTAINED VIA SUBROUTINE 'CXDATA'.
C
C  INPUT : (I*4)  MXTAB   = MAXIMUM NUMBER OF EMISSIVITY TABLES.
C  INPUT : (I*4)  MXGRF   = MAXIMUM NUMBER OF EMISSIVITY GRAPHS.
C  INPUT : (C*2)  SYMBD   = ELEMENT SYMBOL OF DONOR.
C  INPUT : (I*4)  IDZ0    = DONOR NUCLEAR CHARGE.
C  INPUT : (C*2)  SYMBR   = ELEMENT SYMBOL OF RECEIVER.
C  INPUT : (I*4)  IRZ0    = RECEIVER NUCLEAR CHARGE.
C  INPUT : (I*4)  IRZ1    = RECEIVER ION INITIAL CHARGE.
C  INPUT : (I*4)  IRZ2    = RECEIVER ION FINAL CHARGE.
C  INPUT : (I*4)  NGRND   = MINIMUM ALLOWED N QUANTUM NUMBER.
C  INPUT : (I*4)  NTOT    = MAXIMUM ALLOWED N QUANTUM NUMBER.
C
C ROUTINES:
C           ROUTINE     SOURCE      BRIEF DESCRIPTION
C           ------------------------------------------------------------
C
C AUTHOR:  Tim Hammond (TESSELLA SUPPORT SERVICES PLC)
C          RAL EXT. 6404
C
C DATE:    05/06/95
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      CHARACTER*2         SYMBD,       SYMBR
      INTEGER             IDZ0,        IRZ0,        IRZ1,        IRZ2
      INTEGER             MXGRF,       MXTAB,       NGRND,       NTOT
```

4.100 cxsgei: Subroutine cxsgei from library adas3xx

```
      SUBROUTINE CXSGEI( MXNSHL , IZ1    , ZT     , ZT1    , THETA  ,
     &                   VEL    , NA     , LA     , NB     ,
     &                   XSECNA , FRACLA
     &                 )
C
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: CXSGEI *********************
C
C  PURPOSE:  USES THE EIKONAL APPROXIMATION TO CALCULATE CHARGE EXCHANGE
C            CROSS-SECTIONS INTO N'L' RESOLVED EXCITED STATES OF
C            HYDROGENIC IONS IN CAPTURE FROM HYDROGEN NL STATES.
C
C            THE SUBROUTINE CAN BE USED FOR CAPTURE FROM HELIUM BY
C            APPROPRIATE CHOICE OF ZT AND ZT1.
C
C            FOR HYDROGEN, ZT = ZT1 = 1.0
C            FOR HELIUM  , ZT = ZT1 = 1.6875
C
C            EICHLER (1981) PHYS.REV.A,23,498.
C
C  CALLING PROGRAM: ADAS308.
C
C  INPUT : (I*4)  MXNSHL   = MAXIMUM VALUE OF N QUANTUM NUMBER.
C  INPUT : (I*4)  IZ1      = HYDROGENIC ION CHARGE (PROJECTILE).
C  INPUT : (R*8)  ZT       = SCREENING CHARGE FOR THE 1S ELECTRON OF THE
C                            TARGET ATOM IN THE INITIAL STATE.
C  INPUT : (R*8)  ZT1      = EFFECTIVE CHARGE FOR THE 1S ELECTRON OF THE
C                            TARGET ATOM IN THE FINAL STATE.
C  INPUT : (R*8)  THETA    = PARAMETER TO GIVE CORRECT BINDING ENERGY
C                            FOR INITIAL TARGET STATE.
C  INPUT : (R*8)  VEL      = COLLISION SPEED.
C                            UNITS: AT. UNITS
C  INPUT : (I*4)  NA       = PRINCIPAL QUANTUM NUMBER OF INTIAL STATE OF
C                            TARGET FROM WHICH CAPTURE IS MADE.
C  INPUT : (I*4)  LA       = ORBITAL QUANTUM NUMBER OF INTIAL STATE OF
C                            TARGET FROM WHICH CAPTURE IS MADE.
C  INPUT : (I*4)  NB       = PRINCIPAL QUANTUM NUMBER OF FINAL STATES OF
C                            PROJECTILE TO WHICH CAPTURE IS MADE.
C
C  OUTPUT: (R*8)  XSECNA   = N-RESOLVED CROSS-SECTION FOR CAPTURE.
C                            UNITS: AT. UNITS
C  OUTPUT: (R*8)  FRACLA() = L-RESOLVED CROSS-SECTION AS A FRACTION OF
C                            CORRESPONDING N-RESOLVED CROSS-SECTION.
C                            DIMENSION: REFERENCED BY L QUANTUM NUMBER.
C
C  PARAM : (I*4)  MXN      = 'MXNSHL'.
C  PARAM : (R*8)  PI       = PI.
C
C          (I*4)  LB       = ORBITAL QUANTUM NUMBER OF FINAL STATE OF
C                            PROJECTILE.
C          (I*4)  I        = LOOP INDEX.
C          (I*4)  K        = LOOP INDEX.
C          (I*4)  IU       = LOOP INDEX.
C          (I*4)  MB       = LOOP INDEX.
C          (I*4)  MB1      = LOOP INDEX.
C          (I*4)  IND      =
C          (I*4)  IL       = LOOP INDEX.
```

```
C          (I*4)  IG       = LOOP INDEX.
C          (I*4)  IWM      = LOOP INDEX.
C          (I*4)  IW1      = LOOP INDEX.
C          (I*4)  IW2      = LOOP INDEX.
C          (I*4)  IW3      = LOOP INDEX.
C          (I*4)  IW4      = LOOP INDEX.
C          (I*4)  IBA      = LOOP INDEX.
C          (I*4)  IVA      = LOOP INDEX.
C          (I*4)  ITA      = LOOP INDEX.
C          (I*4)  IAM      =
C          (I*4)  IA1      = LOOP INDEX.
C          (I*4)  IA2      = LOOP INDEX.
C          (I*4)  IA3      =
C          (I*4)  ISA      =
C          (I*4)  IBB      = LOOP INDEX.
C          (I*4)  IVB      = LOOP INDEX.
C          (I*4)  ITB      = LOOP INDEX.
C          (I*4)  IAM      =
C          (I*4)  IB1      = LOOP INDEX.
C          (I*4)  IB2      = LOOP INDEX.
C          (I*4)  IB3      =
C          (I*4)  ISB      =
C          (I*4)  IP       = LOOP INDEX.
C
C          (R*8)  Z1       = REAL VALUE = IZ1.
C          (R*8)  XNA      = REAL VALUE = N.
C          (R*8)  XNB      = REAL VALUE = N.
C          (R*8)  ETA      =
C          (R*8)  EPS      =
C          (R*8)  PM       =
C          (R*8)  PP       =
C          (R*8)  QNA      =
C          (R*8)  QNB      =
C          (R*8)  QP       =
C          (R*8)  ZT12     = ZT12**2.
C          (R*8)  T1       =
C          (R*8)  T2       =
C          (R*8)  T3       =
C          (R*8)  T4       =
C          (R*8)  T5       =
C          (R*8)  T6       =
C          (R*8)  T7       =
C          (R*8)  T8       =
C          (R*8)  SUM      =
C          (R*8)  SUM1     =
C          (R*8)  BB       =
C          (R*8)  BB1      =
C
C          (Z*16) CTA      =
C          (Z*16) CTB      =
C          (Z*16) CT1      =
C          (Z*16) CSUM     =
C
C          (R*8)  B()      =
C          (R*8)  F()      =
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4UNIT     ADAS      RETURNS UNIT NO. FOR OUTPUT OF MESSAGES.
C          I4JGAM     ADAS      RETURNS VALUE FROM 'JGAM' TABLE.
```

```
C          R8GAM      ADAS      RETURNS VALUE FROM 'GAM' TABLE.
C          Z16CD      ADAS
C
C AUTHOR:  JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 5183
C
C DATE:    06/10/93
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IZ1,         LA,          MXNSHL,      NA
      INTEGER             NB
      REAL*8              FRACLA(MXNSHL),           THETA,       VEL
      REAL*8              XSECNA,      ZT,          ZT1
```

4.101 cxsode: Subroutine cxsode from library adas3xx

```
      SUBROUTINE CXSODE( IZ0 , N , L , ZEFF , E0 )
C
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: CXSODE ********************
C
C  PURPOSE:  PROVIDES BINDING ENERGY OF TERM CENTRE FOR OUTER ELECTRON
C            IN SODIUM LIKE IONS.
C
C            FROM EDLEN (1978) PHYSICA SCRIPTA, 17, 565.
C
C            FINE STRUCTURE FOR L>0 MUST BE ADDED EXTERNALLY.
C
C            THE ROUTINE IS INCOMPLETE - OPERATES FOR SELECTED IONS
C            ONLY.
C
C  CALLING PROGRAM: GENERAL USE
C
C  INPUT : (I*4)  IZ0     = NUCLEAR CHARGE.
C  INPUT : (I*4)  N       = PRINCIPAL QUANTUM NUMBER.
C  INPUT : (I*4)  L       = ORBITAL QUANTUM NUMBER.
C
C  OUTPUT: (R*8)  ZEFF    = EFFECTIVE CHARGE FOR SPIN-ORBIT SEPAR.
C                           (= ION CHARGE + 1 USUALLY).
C  OUTPUT: (R*8)  E0      = BINDING ENERGY (+VE FOR BOUND STATES).
C                           UNITS: RYD
C
C  PARAM : (I*4)  MXQ     = MAXIMUM ION INDEX FOR QUANTUM DEFECTS DATA.
C  PARAM : (I*4)  MXT     =
C
C          (I*4)  IZ1     = IZ0-10.
C          (I*4)  K       = ARRAY INDEX.
C          (I*4)  IZ1     = LOOP INDEX.
C
C          (R*8)  Z0      = REAL VALUE = IZ0.
C          (R*8)  XN      = REAL VALUE = N.
C          (R*8)  XL      = REAL VALUE = L.
C          (R*8)  R       =
C          (R*8)  A       =
C          (R*8)  B       =
C          (R*8)  C       =
C          (R*8)  S       =
C          (R*8)  U       =
C          (R*8)  U0      =
C          (R*8)  V       =
C          (R*8)  TNL     =
C          (R*8)  AK      =
C          (R*8)  DP      =
C
C          (I*4)  ITAB()  = LOOK UP TABLE FOR QUANTUM DEFECTS DATA.
C                           = 0 : NO DATA PRESENT.
C                           = 1 : DATA PRESENT.
C                           DIMENSION: ION CHARGE INDEX ( = Z+1 ).
C
C          (R*8)  STAB(,) = QUANTUM DEFECTS FOR S.
C                           1ST DIMENSION: 3
C                           2ND DIMENSION: ION CHARGE INDEX ( = Z+1 ).
C          (R*8)  PTAB(,) = QUANTUM DEFECTS FOR P.
```

```
C                           1ST DIMENSION: 3
C                           2ND DIMENSION: ION CHARGE INDEX ( = Z+1 ).
C          (R*8)  DTAB(,) = QUANTUM DEFECTS FOR D.
C                           1ST DIMENSION: 3
C                           2ND DIMENSION: ION CHARGE INDEX ( = Z+1 ).
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4UNIT     ADAS      RETURNS UNIT NO. FOR OUTPUT OF MESSAGES.
C          R8P        ADAS
C          R8QP       ADAS
C
C AUTHOR:  JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 5183
C
C DATE:    03/11/93
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IZ0,         L,           N
      REAL*8              E0,          ZEFF
```

4.102 cxsqef: Subroutine cxsqef from library adas3xx

```
      subroutine cxsqef( iunit  , dsname , ibsel  ,
     &                   nqeff  , epro   , ttar   ,
     &                   em1    , em2    , iord   ,
     &                   ti     , densi  , zeff   , bmag   ,
     &                   nener  , ener   , qener  ,
     &                   csymb  , czion  , cwavel , cdonor , crecvr  ,
     &                   ctrans , cfile  , ctype  , cindm  ,
     &                   qeff   , ircode
     &                 )
C-----------------------------------------------------------------------
C
C ******************** FORTRAN77 SUBROUTINE : CXSQEF *****************
C
C  PURPOSE:  Subroutine to evaluate Maxwell averaged effective rate
C            coefficients for charge exchange/Stark studies.
C
C            The source data is effective coefficients in the
C            collisional/radiative sense or effective emission
C            coefficients for photon emission but before averaging
C            over ion/atom speed distribution functions.
C
C            The function also returns the raw eff. coefft. data for
C            verification and graphing purposes.
C
C            The effective rate coefficient appropriate to one of
C            the particles being in a monoenergetic beam and the other
C            belonging to a Maxwell distribution may be returned.
C            The target and projectile roles may be reversed. Arbitrary
C            relative speeds are allowed.
C
C  SUBROUTINE:
C
C  input : (i*4) iunit   = unit number on which ionatom file is opened
C  input : (c)   dsname  = full name of data set to be opened and read
C  input : (i*4) ibsel   = selector for particular rate coefft.
C
C  input : (i*4) nqeff   = number of rates to be evaluated (when ttar>0)
C                          a 1d array of plasma/beam conditions are
C                          evaluated to give a vector of rates. at
C                          the moment, epro, ttar, ti, densi, zeff &
C                          bmag are allowed to vary along the vector.
C  input : (r*8) epro    = incident particle energy (ev/amu)
C  input : (r*8) ttar    = maxwell temperature of target particles (ev)
C                          if (ttar.le.0) then rates for t=0 are
C                          returned
C  input : (r*8) em1     = atomic mass number of first particle
C  input : (r*8) em2     = atomic mass number of second particle
C  input : (i*4) iord    = 1 for 1st particle incident and monoenergetic
C                          = 2 for 2nd particle incident and monoenergetic
C  input : (r*8) ti      = plasma ion temperature (ev)
C  input : (r*8) densi   = plasma ion density (cm-3)
C  input : (r*8) zeff    = plasma z effective
C  input : (r*8) bmag    = plasma magnetic field (tesla)
C
C  output: (r*8) qeff    = rate coefficient (cm3 sec-1)
C  output: (i*4) nener   = number of source data values
C  output: (r*8) ener(i) = set of energies (ev/amu) for
C                          selected source data
C  output: (r*8) qener(i)= rate coeffts.(cm**3 sec-1) for
```

```
C                           selected source data
C  output: (c*2) csymb   = element symbol
C  output: (c*3) czion   = emitting ion charge
C  output: (c*8) cwavel  = wavelength (A)
C  output: (c*6) cdonor  = donor neutral atom
C  output: (c*5) crecvr  = receiver nucleus
C  output: (c*7) ctrans  = transition
C  output: (c*10)cfile   = specific ion file source
C  output: (c*2) ctype   = type of emissivity
C  output: (c*3) cindm   = emissivity index
C  output: (i*4) ircode  = return code from subroutine:
C                           0 => normal completion - no error detected
C                           1 => error opening requested data set
C                                exist - data set not connected
C                           3 => the selected data-block 'ibsel' is out
C                                of range or does not exist.
C
C
C  ROUTINES:
C           ROUTINE    SOURCE    BRIEF DESCRIPTION
C           ------------------------------------------------------------
C           xxdata_12   ADAS     reads values from 'ionatom' dataset
C           c3corr      ADAS     calculates scaled plasma parameter
C           c3alrs      ADAS     calculates rate coefficient
C
C  AUTHOR:  C. J. WHITEHEAD,  UNIVERSITY OF STRATHCLYDE
C
C  DATE:    25/11/94
C
C  UPDATE:  19/12/94  HP SUMMERS - TIDIED UP FORMATTING
C           03/01/95  HP SUMMERS - CORRECTED THERMAL AVERAGED RATE
C                                  COEFFICIENT BY INTRODUCING OAA ARRAY
C
C  UPDATE:  11/01/95  PE BRIDEN  - CHANGED DSNAME FROM C*30 TO C*44
C                                  TO AGREE WITH THAT IN C2FILE.
C                                - INITIALISE NBSEL AS ZERO.
C
C  UPDATE:  03/05/95  PE BRIDEN  - C3DATA CHANGED TO C3DATAO AS CURRENT
C                                  VERSION OF SQEF NEEDS TTO BE UPDATED
C                                  TO USE THE NEW VERSION OF C3DATA.
C
C  UPDATE:  15/05/95  Tim Hammond - UNIX PORT
C                                   Put under SCCS control
C-----------------------------------------------------------------------
C
C  Copied from ...adas3xx/adas303/sqef.for, renamed and relocated as
C  ...adas3xx/adaslib/cxsqef.for.
C
C  VERSION  : 1.1
C  DATE     : 15-11-2002
C  MODIFIED : Lorne Horton
C               - First version
C               - Switched to ADAS-standard C3DATA.
C                 This is primarily a change to requiring the full
C                 input file name as input.
C               - Increased NSTORE to 150 - consistent with ADAS303
C               - Added loop to allow multiple evaluations per call.
C                 This means changing from a function to a
C                 subroutine
C               - Removed IPASS. Routine now re-reads data sets only
C                 when theinput name has changed.
```

```
C               - Added SAVE statement
C
C
C  VERSION  : 1.2
C  DATE     : 02-12-2004
C  MODIFIED : Martin O'Mullane
C               - Replace c3data with xxdata_12.
C               - Place into central ADAS.
C
C  VERSION  : 1.3
C  DATE     : 17-05-2007
C  MODIFIED : Allan Whiteford
C               - Updated comments as part of subroutine documentation
C                 procedure.
C
C  VERSION  : 1.4
C  DATE     : 05-06-2007
C  MODIFIED : Martin O'Mullane
C               - New version of xxdata_12 with extra outputs.
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      CHARACTER*6         CDONOR
      CHARACTER*10        CFILE
      CHARACTER*3         CINDM
      CHARACTER*5         CRECVR
      CHARACTER*2         CSYMB
      CHARACTER*7         CTRANS
      CHARACTER*2         CTYPE
      CHARACTER*8         CWAVEL
      CHARACTER*2         CZION
      CHARACTER*132       DSNAME
      INTEGER             IBSEL,        IORD,         IRCODE,      IUNIT
      INTEGER             NENER,        NQEFF
      REAL*8              BMAG(NQEFF), DENSI(NQEFF),              EM1
      REAL*8              EM2,          ENER(MENER), EPRO(NQEFF)
      REAL*8              QEFF(NQEFF), QENER(MENER,NQEFF)
      REAL*8              TI(NQEFF),    TTAR(NQEFF), ZEFF(NQEFF)
```

4.103 cxtbex: Subroutine cxtbex from library adas3xx

```
      SUBROUTINE CXTBEX( MXNSHL , IZ1    , NBOT   , NTOP   ,
     &                   NGRND  , TEV    , TBQEX  , QTHEX  ,
     &                   FTHEX
     &                 )
C
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: CXTBEX *********************
C
C  PURPOSE:  SETS UP A TABLE OF ELECTRON IMPACT EXCITATION RATE
C            COEFFICIENTS FOR A HYDROGENIC ION FROM THE GROUND STATE
C            TO EXCITED NL-LEVELS.
C
C  CALLING PROGRAM: ADAS308 , C6TBEX.
C
C  INPUT : (I*4)  MXNSHL   = MAXIMUM VALUE OF N QUANTUM NUMBER.
C  INPUT : (I*4)  IZ1      = ION CHARGE.
C  INPUT : (I*4)  NBOT     = MINIMUM PRINCIPAL QUANTUM NUMBER.
C  INPUT : (I*4)  NTOP     = MAXIMUM PRINCIPAL QUANTUM NUMBER.
C  INPUT : (I*4)  NGRND    =
C  INPUT : (R*8)  TEV      = ELECTRON TEMPERATURE.
C                            UNITS: EV
C
C  OUTPUT: (R*8)  TBQEX()  = TABLE OF NL-LEVEL EXCITATION RATE
C                            COEFFICIENTS.
C                            UNITS:
C                            DIMENSION: REFERENCED BY FUNC I4IDFL(N,L).
C  OUTPUT: (R*8)  QTHEX()  = TABLE OF N-LEVEL EXCITATION RATE
C                            COEFFICIENTS.
C                            UNITS:
C                            DIMENSION: N-SHELL
C  OUTPUT: (R*8)  FTHEX()  = TABLE OF NL-LEVEL EXCITATION RATE
C                            COEFFICIENTS EXPRESSED AS FRACTION OF
C                            CORRESPONDING N-LEVEL RATE.
C                            DIMENSION: REFERENCED BY FUNC I4IDFL(N,L).
C
C  PARAM : (I*4)  MXN      = 'MXNSHL'.
C  PARAM : (R*8)  P1       =
C
C          (I*4)  N        = N QUANTUM NUMBER.
C          (I*4)  L        = L QUANTUM NUMBER.
C          (I*4)  IDL      = L-RESOLVED TABLE INDEX.
C
C          (R*8)  ATE      =
C          (R*8)  RDE      =
C          (R*8)  ETE      =
C          (R*8)  FACT     =
C
C          (R*8)  GAMA()   = TABLE OF EXCITATION RATE PARAMETERS.
C                            UNITS:
C                            DIMENSION: REFERENCED BY N QUANTUM NUMBER.
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4UNIT     ADAS      RETURNS UNIT NO. FOR OUTPUT OF MESSAGES.
C          I4IDFL     ADAS      RETURNS UNIQUE INDEX GIVEN QUANTUM
C                               NUMBERS N AND L.
```

```
C          CXGHNL     ADAS      CALCULATES EXCITATION RATE PARAMETERS.
C
C AUTHOR:   JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C           K1/0/81
C           JET EXT. 5183
C
C DATE:     05/10/93
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IZ1,         MXNSHL,      NBOT,        NGRND
      INTEGER             NTOP
      REAL*8              FTHEX((MXNSHL*(MXNSHL+1))/2)
      REAL*8              QTHEX(MXNSHL)
      REAL*8              TBQEX((MXNSHL*(MXNSHL+1))/2),          TEV
```

4.104 cxtblf: Subroutine cxtblf from library adas3xx

```
      SUBROUTINE CXTBLF( MXNSHL , IZ1 , NBOT , NTOP , TBLF )
C
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: CXTBLF *********************
C
C  PURPOSE: SETS UP A RADIATIVE LIFETIME TABLE FOR NL LEVELS OF A
C           HYDROGENIC ION.
C
C  CALLING PROGRAM: ADAS306 , ADAS308.
C
C  INPUT : (I*4)  MXNSHLL  = MAXIMUM VALUE OF N QUANTUM NUMBER.
C  INPUT : (I*4)  IZ1      = ION CHARGE.
C  INPUT : (I*4)  NBOT     = MINIMUM PRINCIPAL QUANTUM NUMBER.
C  INPUT : (I*4)  NTOP     = MAXIMUM PRINCIPAL QUANTUM NUMBER.
C
C  OUTPUT: (R*8)  TBLF()   = TABLE OF RADIATIVE LIFETIMES.
C                            UNITS: SECS
C                            DIMENSION: REFERENCED BY FUNC I4IDFL(N,L).
C
C          (I*4)  NU       = UPPER VALUE OF N QUANTUM NUMBER.
C          (I*4)  LU       = L QUANTUM NUMBER FOR NU.
C          (I*4)  NL       = LOWER VALUE OF N QUANTUM NUMBER.
C          (I*4)  LL       = L QUANTUM NUMBER FOR NL.
C          (I*4)  IDL      = TABLE INDEX.
C
C          (R*8)  SUM     = SUM OF A-VALUES FOR GIVEN NU AND LU.
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          R8ATAB     ADAS      RETURNS HYDRONIC L-RESOLVED A-VALUES.
C          I4IDFL     ADAS      RETURNS UNIQUE INDEX FROM QUANTUM
C                               NUMBERS N AND L.
C
C AUTHOR:   JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C           K1/0/81
C           JET EXT. 5183
C
C DATE:     30/09/93
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IZ1,         MXNSHL,      NBOT,        NTOP
      REAL*8              TBLF((MXNSHL*(MXNSHL+1))/2)
```

4.105 cxther: Subroutine cxther from library adas3xx

```
C
       SUBROUTINE CXTHER ( NEDIM  , NTDIM  ,
     &                     LSETX  , LPASS  ,
     &                     AMDON  , AMREC  ,
     &                     ALPH   , ETH    , ILTYP  ,
     &                     NENIN  , ENIN   , NENOUT , ENOUT ,
     &                     SGIN   , RCOUT
     &                   )
C-----------------------------------------------------------------------
C
C  ********************** FORTRAN77 SUBROUTINE: CXTHER *****************
C
C  VERSION: 1.0 (ADAS91)
C
C  PURPOSE:  OBTAINS RATE COEFFICIENTS FOR DONOR/RECEIVER CHARGE
C            EXCHANGE COLLISIONS FOR CASES OF THERMAL DONOR AND
C            THERMAL RECEIVER FROM CROSS-SECTION TABULATIONS. AN ARRAY
C            OF VALUES IS PRODUCED.
C
C  CALLING PROGRAM:  ADAS302
C
C  SUBROUTINE:
C
C  INPUT :  (I*4)  NEDIM    = MAX. NUMBER OF ENERGIES IN SOURCE DATA
C                             VECTOR
C  INPUT :  (I*4)  NTDIM    = MAX. NUMBER OF TEMPERATURES IN OUTPUT
C                             VECTOR
C  INPUT :  (L*4)  LSETX    = .TRUE. => SPLINE PRESET FOR THESE KNOTS
C                             .FLSE. => SPLINE NOT SET FOR THESE KNOTS
C  INPUT :  (L*4)  LPASS    = .TRUE. => DO NOT CONVERT INTO LOG10 FOR
C                                        ENERGIES AND X-SECTS. FOR SPLINE
C                             .FLSE. => CONVERT INTO LOG10 FOR
C                                        ENERGIES AND X-SECTS. FOR SPLINE
C  INPUT :  (R*8)  AMDON    = DONOR MASS NUMBER
C  INPUT :  (R*8)  AMREC    = RECEIVER MASS NUMBER
C  INPUT :  (R*8)  ALPH     = HIGH ENERGY EXTENSION PARAMETER
C  INPUT :  (R*8)  ETH      = THRESHOLD ENERGY (RYD.)
C  INPUT : (I*4)   ILTYP    = TYPE FOR LOW AND HIGH ENERGY CROSS-
C                             SECTION EXTRAPOLATION
C
C  INPUT : (I*4)   NENIN    = NUMBER OF ENERGIES IN INPUT DATA SET
C  INPUT : (R*8)   ENIN()   = ENERGIES (EV/AMU) IN INPUT DATA SET
C  INPUT : (I*4)   NENOUT   = NUMBER OF TEMPERATURES FOR OUTPUT DATA SET
C  INPUT : (R*8)   ENOUT()  = TEMPERATURES (EV) FOR OUTPUT DATA SET
C  INPUT : (R*8)   SGIN()   = INPUT X-SECTIONS (CM2) FROM INPUT DATA SET
C                             1ST.DIM: ENERGY INDEX
C  OUTPUT: (R*8)   RCOUT(,) = RATE COEFF. (CM3 S-1) IN OUTPUT DATA SET
C                             1ST.DIM: DONOR TEMPERATURE INDEX
C                             2ND.DIM: RECEIVER TEMPERATURE INDEX
C
C          (I*4)   I        = GENERAL INDEX
C          (I*4)   IT       = GENERAL INDEX
C          (I*4)   ITR      = GENERAL INDEX
C          (I*4)   ITD      = GENERAL INDEX
C          (I*4)   ITHETA   = GENERAL INDEX
C          (I*4)   IOPT     = SPLINE END POINT CURVATURE/GRADIENT OPTION
C                             1 => DDY1 = 0, DDYN = 0
C                             4 => DY1 = 0 , DDYN = 0
C          (I*4)   IXD      = DONOR GAUSSIAN QUADRATURE INDEX
```

```
C          (I*4)   IXR      = RECEIVER GAUSSIAN QUADRATURE INDEX
C          (I*4)   NGS      = GAUSSIAN QUARATURE DIMENSION
C          (I*4)   NTHETA   = NUMBER OF ANGLE VALUES FOR QUADRATURE
C          (I*4)   LTHETA   = NTHETA+1
C          (I*4)   L1       = PARAMETER = 1
C
C          (R*8)   ETHD     = THERMAL ENERGY OF DONOR       (JOULES)
C          (R*8)   ETHR     = THERMAL ENERGY RECEIVER       (JOULES)
C          (R*8)   HSIMP    = SIMPSON'S RULE STEP INTERVAL
C          (R*8)   THETA    = ANGLE BETWEEN PARTICLE VELOCITIES (RAD)
C          (R*8)   FAC      = GENERAL VARIABLE
C          (R*8)   FLAG     = GENERAL VARIABLE
C          (R*8)   XMDKG    = DONOR MASS    (KG)
C          (R*8)   XMRKG    = RECEIVER MASS (KG)
C          (R*8)   VD       = DONOR SPEED    (M S-1)
C          (R*8)   VR       = RECEIVER SPEED (M S-1)
C          (R*8)   RATE     = EVALUATED RATE COEFFICIENT (CM3 S-1)
C          (R*8)   PART1    = GENERAL VARIABLE
C          (R*8)   PART2    = GENERAL VARIABLE
C          (R*8)   PART3    = GENERAL VARIABLE
C          (R*8)   PART12   = GENERAL VARIABLE
C          (R*8)   PART23   = GENERAL VARIABLE
C          (R*8)   PART123  = GENERAL VARIABLE
C          (R*8)   VREL1    = GENERAL RELATIVE SPEED VARIABLE
C          (R*8)   XSEC1    = GENERAL CROSS-SECTION VARIABLE
C          (R*8)   VAL      = GENERAL VARIABLE
C
C          (R*8)   XGS()    = GAUSSIAN QUADRATURE NODES
C          (R*8)   WGS()    = GAUSSIAN QUADRATURE WEIGHTS
C          (R*8)   VREL()   = RELATIVE SPEED OF PARTICLES FOR DIFFERENT
C                             ANGLES (CM S-1)
C          (R*8)   XSEC()   = CHARGE EXCHANGE CROSS-SECTIONS FOR
C                             RELATIVE SPEEDS AT DIFFERENT ANGLES (CM2)
C
C
C  ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          SIGCX      ADAS      INTERPOLATES CX CROSS-SECTION TABLES
C
C
C  AUTHOR:  H. P. SUMMERS, UNIVERSITY OF STRATHCLYDE
C           JA8.08
C           TEL. 0141-553-4196
C
C  DATE:    17/11/95
C
C UNIX-IDL PORT: H.P.SUMMERS
C
C VERSION: 1.1 DATE: 19-11-96
C MODIFIED: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C     - PUT UNDER S.C.C.S. CONTROL
C
C-----------------------------------------------------------------------
      INTEGER             ILTYP,       NEDIM,       NENIN,       NENOUT
      INTEGER             NTDIM
      LOGICAL             LPASS,       LSETX
      REAL*8              ALPH,        AMDON,       AMREC
      REAL*8              ENIN(NEDIM), ENOUT(NTDIM),             ETH
      REAL*8              RCOUT(NTDIM,NTDIM),       SGIN(NEDIM)
```

4.106 diel: Subroutine diel from library adas3xx

```
       SUBROUTINE DIEL(Z,EIJ,F,T,COR,JCOR,N,AD)
       IMPLICIT REAL*8 (A-H,O-Z)
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 PROGRAM: DIEL *************************
C
C PURPOSE UNKNOWN
C
C NOTES: THIS ROUTINE IS NOT YET PROPERLY ANNOTATED
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 16-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST VERSION
C
C-----------------------------------------------------------------------
C
      INTEGER             JCOR,        N
      REAL*8              AD,          COR(20),     EIJ,         F
      REAL*8              T,           Z
```

4.107 diel_310: Subroutine diel_310 from library adas3xx

```
       SUBROUTINE DIEL_310(Z,EIJ,F,T,COR,JCOR,N,AD)
       IMPLICIT REAL*8 (A-H,O-Z)
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 PROGRAM: DIEL_310 *************************
C
C PURPOSE UNKNOWN
C
C NOTES: THIS ROUTINE IS NOT YET PROPERLY ANNOTATED
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 16-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST VERSION
C
C VERSION: 1.2 DATE: 04-04-2000
C MODIFIED: RICHARD MARTIN
C   - CHANGED NAME FROM DIEL.FOR TO DIEL_310.FOR
C     (TO AVOID NAME CONFLICT WITH ADAS801)
C
C-----------------------------------------------------------------------
C
      INTEGER             JCOR,        N
      REAL*8              AD,          COR(20),     EIJ,         F
      REAL*8              T,           Z
```

4.108 drv: Subroutine drv from library adas3xx

```
      SUBROUTINE DRV
      IMPLICIT REAL*8 (A-H,O-Z)
C-----------------------------------------------------------------------
C
C     PURPOSE: Calculates coefficients for spline approximation
C              of tabulated function in 10 points together with
C              subroutine DRVSPL.
C
C-----------------------------------------------------------------------
C
C ADAS305 version. Developed from JETSHP.STARK.FORT (H P Summers).
C
C VERSION  : 1.1
C DATE     : 24-02-2005
C MODIFIED : Martin O'Mullane
C             - First version.
C
C VERSION  : 1.2
C DATE     : 16-05-2007
C MODIFIED : Allan Whiteford
C             - Updated comments as part of subroutine documentation
C               procedure.
C
C-----------------------------------------------------------------------
      COMMON/APRX/F(10),B(10),H(10),ABETA(10)
```

4.109 eiqip: Subroutine eiqip from library adas3xx

```
       SUBROUTINE EIQIP(EI,EIJ,EM,Z,PHI,SC,WI,WJ,R,EIQ,FLAG)
C
       IMPLICIT REAL*8 (A-H,O-Z)
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: EIQIP **********************
C
C PURPOSE UNKNOWN
C
C NOTES: THIS ROUTINE IS NOT YET PROPERLY ANNOTATED
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 16-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST VERSION
C
C-----------------------------------------------------------------------
C
       FLAG=1.0
       EF=EI-EIJ
       T=EF/EI
       C=1.0
       IF(T-0.2)1,4,4
    1  IF(Z-0.01)2,3,3
    2  C=DSQRT(4.0*EF/EIJ)
       GO TO 21
    3  IF(EM-1.5)21,21,20
   20     C=DABS(EIJ)/(EI*DSQRT(EF)+EF*DSQRT(EI))
       C=Z*DSQRT(EM)*(1.1056/DSQRT(EIJ)-C)
       C=DEXP(3.142*C)
   21     EI=1.25*EIJ
       EF=0.25*EIJ
       GO TO 8
    4  IF(T-5.0)8,8,5
    5  IF(Z-0.01)6,6,7
    6  C=DSQRT(-4.0*EI/EIJ)
       GO TO 23
    7  IF(EM-1.5)23,23,22
   22     C=DABS(EIJ)/(EI*DSQRT(EF)+EF*DSQRT(EI))
       C=Z*DSQRT(EM)*(1.1056/DSQRT(-EIJ)-C)
       C=DEXP(3.142*C)
   23     EI=-0.25*EIJ
       EF=-1.25*EIJ
    8  TI=DSQRT(EI)
       TF=DSQRT(EF)
       TIF=DABS(EIJ)/(TI+TF)
       XI=Z*TIF/(TI*TF)
       D=TIF*R
       E=TI*TF
       IF(EM-1.5)25,24,24
   24  EM2=DSQRT(EM)
          TIF=EM2*TIF
          XI=-EM2*XI
          D=EM2*D
   25  CONTINUE
       T=Z+E*R
       T1=XIP(XI,D)/(T*T)
```

```
      T2=4.0*PHI*E*SC*EM
      IF(WI-WJ)10,10,9
    9 T2=T2*WI/WJ
   10 P=T2*T1
      EIQW=8.0*C*PHI*YIP(XI,D)*EM
      IF(P-0.5)15,15,11
   11 R1=R
   12 A=R1
      FLAG=0.0
      VA=0.5-P
      R1=R1+R1
      T=Z+E*R1
      D=TIF*R1
      P=T2*XIP(XI,D)/(T*T)
      IF(P-0.5)13,14,12
   13 B=R1
      VB=0.5-P
      CALL ZERO1(A,B,VA,VB,0.01D0,R1,XI,Z,E,TIF,T2)
   14 T=Z+E*R1
      D=TIF*R1
      T1=XIP(XI,D)/(T*T)
   15 EIQ=8.0*C*PHI*(YIP(XI,D)+0.5*(T*T-Z*Z)*T1)*EM
      EIQ=DMIN1(EIQ,EIQW)
      RETURN
      END
     REAL*8              EI,           EIJ,          EIQ,          EM
     REAL*8              FLAG,         PHI,          R,            SC
     REAL*8              WI,           WJ,           Z
```

4.110 find: Subroutine find from library adas3xx

```
      SUBROUTINE FIND(ARR,VALUE,IMAX,INDEX)

C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: FIND *********************
C
C  PURPOSE:
C  SUBROUTINE TO FIND VALUES FROM LINE AND STORE ONLY NOT REPEATED
C  VALUES
C
C
C  INPUT
C       ARR = NAME OF ARRAY TO STORE ELEMENTS
C       VALUE = VALUE TO BE FOUND IN LINE
C       IMAX = TOTAL NUMBER OF NOT REPEATED ELEMENTS IN ARRAY
C       INDEX = COUNTER OF ELEMENT POSITION IN ARRAY
C  OUTPUT
C        ARR = ARRAY CONTAINING NOT REPEATED VALUES READ FROM LINE
C
C NOTES: THIS ROUTINE IS NOT YET PROPERLY ANNOTATED
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 01-02-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - PUT UNDER SCCS CONTROL
C
C-----------------------------------------------------------------------
      LOGICAL OVER,FOUND
      INTEGER             IMAX,         INDEX
      REAL                ARR(*),       VALUE
```

4.111 finish5: Subroutine finish5 from library adas3xx

```
         SUBROUTINE FINISH5(NIP,INTD,IPRS,ILOW,IONIP,NIONIP,ILPRS,IVDISP,
     &       ZEFF,TS,W,CION,CPY,W1,ZIMPA,DNIMPA,NIMP,IUPS1,IUPS2,STITLE,
     &       NBENG,NTEMP,NDENS, lbndl, lproj)

        IMPLICIT REAL*8(A-H,O-Z)
C-----------------------------------------------------------------------
C
C       ********** FORTRAN 77 ROUTINE : FINISH5.F **********
C
C   PURPOSE : ASSEMBLES AND SOLVES THE COLLISIONAL RADIATIVE
C             FOR THE RELATIVE POPULATIONS, Bnl FACTORS AND THE
C             COLLISIONAL-RADIATIVE IONISATION AND RECOMBINATION
C             COEFFICIENTS.
C
C   INPUT :
C
C           ROUTINE SET TO READ STREAM 12 AS A TEMPORARY MEASURE
C
C   OUTPUT:
C
C
C   HISTORY : ROUTINE WAS ORIGINALLY WRITTEN BY H.P. SUMMERS
C
C
C   NOTE    :
C
C             IPOSNT .EQ. 1 EXTERNAL RADIATION FIELD IS .NE. 0.0
C             THE C-R MATRIX IS MODIFIED AND THE Bnl SOLUTION IS
C             OBTAINED. THE F2 COLUMN REPRESENTING THE RECOMBINATION
C             IS THEN EQUAL TO THE Bnl SOLUTION.
C
C             IPOSNT .EQ. 2 EXTERNAL RADIATION FIELD IS SET TO 0.0
C             PROVIDING THAT IR.EQ.2 AND JR.EQ.1. THE C-R MATRIX IS
C             MODIFIED AND THE Bnl SOLUTION IS OBTAINED. THE F1I
C             COLUMN REPRESENTING THE EXCITATION CONTRIBUTION FROM
C             THE FIRST METASTABLE IS EVALUATED BY MANIPULATING
C             THE F2 COLUMN, Bnl SOLUTION AND THE RELATIVE POPULATION
C             OF THE GROUND STATE.
C
C             IPOSNT .EQ.3 EXTERNAL RADIATION FIELD IS SET TO 0.0
C             PROVIDING THAT IR.EQ.2 AND JR.EQ.1. THE C-R MATRIX IS
C             MODIFIED AND THE Bnl SOLUTION IS OBTAINED. THE F1II
C             COLUMN REPRESENTING THE EXCITATION CONTRIBUTION FROM
C             THE SECOND METASTABLE IS EVALUATED BY MANIPULATING
C             THE F2 COLUMN, Bnl SOLUTION AND THE RELATIVE POPULATION
C             OF THE SECOND METASTABLE STATE.
C
C             IPOSNT .EQ.4 EXTERNAL RADIATION FIELD IS SET TO 0.0
C             PROVIDING THAT IR.EQ.2 AND JR.EQ.IMAX+1. THE C-R MATRIX
C             IS MODIFIED AND THE Bnl SOLUTION IS OBTAINED.THE F1III
C             COLUMN REPRESENTING THE EXCITATION CONTRIBUTION FROM
C             THE SECOND METASTABLE IS EVALUATED BY MANIPULATING
C             THE F2 COLUMN, Bnl SOLUTION AND THE RELATIVE POPULATION
C             OF THE SECOND METASTABLE STATE.
C
C
C
C   CONTACT : HARVEY ANDERSON
C             UNIVERSITY OF STRATHCLYDE
```

```
C             ANDERSON@PHYS.STRATH.AC.UK
C
C   DATE    : 4/3/98
C
C
C VERSION : 1.2
C DATE    : 21-10-99
C MODIFIED: RICHARD MARTIN
C            - CHANGED HEXADECIMAL CONSTANTS TO Z'FFF00000' FORM.
C
C VERSION : 1.3
C DATE    : 3-6-2000
C MODIFIED: Martin O'Mullane
C            - Removed call to errset.
C
C VERSION : 1.4
C DATE    : 18-11-2004
C MODIFIED: Martin O'Mullane
C            - Align with Harvey Anderson's last version.
C            - Add lbndl if adf26 files output is requested.
C            - The dsnps1 variable is replaced by iups2 in the
C              parameter list.
C            - Outputs projection matrices for Vienna codes
C              if lproj is set.
C
C-----------------------------------------------------------------------


      CHARACTER*80        STITLE
      INTEGER             ILOW,         ILPRS,        INTD,         IONIP
      INTEGER             IPRS,         IUPS1,        IUPS2,        IVDISP
      INTEGER             NBENG,        NDENS,        NIMP,         NIONIP
      INTEGER             NIP,          NTEMP
      LOGICAL             LBNDL,        LPROJ
      REAL*8              CION,         CPY,          DNIMPA(10),   TS
      REAL*8              W,            W1,           ZEFF
      REAL*8              ZIMPA(10)
```

4.112 finter: Subroutine finter from library adas3xx

```
       FUNCTION FINTER(EN)
       IMPLICIT REAL*8 (A-H,O-Z)
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 FUNCTION: FINTER ***********************
C
C  PURPOSE:  UNKNOWN
C
C NOTES: THIS ROUTINE IS NOT YET PROPERLY ANNOTATED
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 16-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST VERSION
C
C-----------------------------------------------------------------------
C
      FINTER=1.0/((EN+10.0)*(EN+10.0))
      RETURN
      END
      REAL*8              EN
```

4.113 fitsp: Subroutine fitsp from library adas3xx

```
       SUBROUTINE FITSP(X,XA,N,YAA,Y,DY,I0,C1,C2,C3,C4,ISW)
C
       IMPLICIT REAL*8(A-H,O-Z)
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: FITSP **********************
C
C PURPOSE UNKNOWN
C
C NOTES: THIS ROUTINE IS NOT YET PROPERLY ANNOTATED
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 18-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - PUT UNDER SCCS CONTROL
C
C-----------------------------------------------------------------------
C
      INTEGER             I0,           ISW,          N
      REAL*8              C1(10,9),     C2(10,9),     C3(10,9)
      REAL*8              C4(10,9),     DY,           X,            XA(10)
      REAL*8              Y,            YAA(10)
```

4.114 fsplin: Subroutine fsplin from library adas3xx

```
      SUBROUTINE FSPLIN(N,X,F,B,H,M,XNEW,FSPL)
      IMPLICIT REAL*8 (A-H,O-Z)
C-----------------------------------------------------------------------
C     PURPOSE: Calculates M values of function FSPL(XNEW)
C              using spline approximation
C-----------------------------------------------------------------------
C
C ADAS305 version. Developed from JETSHP.STARK.FORT (H P Summers).
C
C VERSION  : 1.1
C DATE     : 24-02-2005
C MODIFIED : Martin O'Mullane
C             - First version.
C
C VERSION  : 1.2
C DATE     : 16-05-2007
C MODIFIED : Allan Whiteford
C             - Updated comments as part of subroutine documentation
C               procedure.
C
C-----------------------------------------------------------------------
      INTEGER             M,           N
      REAL*8              B(N),        F(N),        FSPL(M),     H(N)
      REAL*8              X(N),        XNEW(M)
```

4.115 gbb: .

]Subroutine gbb from library adas3xx

```
       FUNCTION GBB(EN,EN1,X2,X)
C-----------------------------------------------------------------------
C
C       PURPOSE:    TO EVALUATE THE BOUND BOUND GAUNT FACTOR USING THE
C                   EXPRESSION OUTLINED BY BURGESS.A AND SUMMERS.H.P [1].
C
C       BACKGROUND: THE BOUND BOUND GAUNT FACTOR IS USED IN THE
C                   EVALUATION OF THE EINSTEIN A COEFFICIENT. THE
C                   EINSTEIN A COEFFICIENT IS USUALLY EXPRESSED IN
C                   TERMS OF THE UPWARD OSCILLATOR STRENGTH [2].
C                   HOWEVER,IT IS MORE CONVIENIENT TO WRITE THE
C                   UPWARD OSCILLATOR STRENGTH IN TERMS OF WHAT IS
C                   DESCRIBED AS AN APPROXIMATE UPWARD OSCILLATOR
C                   STRENGTH.THE BOUND BOUND GAUNT FACTOR IS A
C                   CORRECTION TO THE APPROXIMATE OSCILLATOR
C                   STRENGTH [3].
C
C       REFERENCES:
C                   [1] BURGESS.A & SUMMERS.H.P
C                       THE RECOMBINATION AND LEVEL POPULATION OF IONS I
C                       Mon.Not.R.astr.Soc.(1976),174,PP345-391.
C                       ( SEE EQUATION A5 ).
C
C                   [2] SPENCE.J
C                       STUDIES OF CHARGE EXCHANGE RECOMBINATION IN
C                       LABORATORY FUSION PLASMAS.
C                       Phd THESIS.
C                       ( SEE P36 EQUATIONS 2.4.2.1 TO 2.4.2.3 ).
C
C                   [3] MENZEL.D.H & PEKERIS.C.L
C                       Mon.Not.R.astr.Soc.(1935),96,P77.
C
C       CALLING PROGRAM:
C
C       FUNCTION:
C
C       INPUT : (R*8) X2        = THE RECIPROCAL OF THE SQUARE OF THE
C                                 UPPER PRINCIPAL QUANTUM NUMBER.
C       INPUT : (R*8) X         = GENERAL VARIABLE RELATED TO THE
C                                 UPPER PRINCIPAL QUANTUM NUMBER.
C   X=EN23(N)*EN23(N2)/EN23(N11-1)
C       INPUT : (R*4) EN        = THE UPPER PRINCIPAL QUANTUM NUMBER.
C       INPUT : (R*4) EN1       = THE LOWER PRINCIPAL QUANTUM NUMBER.
C
C       OUTPUT: (R*8) GBB       = THE BOUND BOUND GAUNT FACTOR.
C
C               (R*8) G1        = GENERAL VARIABLE.
C               (R*8) G2        = GENERAL VARIABLE.
C               (R*8) G3        = GENERAL VARIABLE.
C               (R*8) T1        = GENERAL VARIABLE.
C               (R*8) T2        = GENERAL VARIABLE.
C               (R*8) T3        = GENERAL VARIABLE.
C               (R*8) T4        = GENERAL VARIABLE.
C               (R*8) X4        = THE RECIPROCAL OF THE PRINCIPAL
C                                 QUANTUM NUMBER CUBED.
C
C
C       CONTACT: HARVEY ANDERSON
```

```
C               UNIVERSITY OF STRATHCLYDE.
C               ROOM 4.13B EXT 4213
C
C       DATE:   23/12/95
C
C
C VERSION: 1.2 DATE: 17-03-99
C MODIFIED: HARVEY ANDERSON
C - ADDED DOCUMENTATION
C
C-----------------------------------------------------------------------
C
      REAL*8              EN,          EN1,         X,           X2
```

4.116 gbf: Subroutine gbf from library adas3xx

```
       FUNCTION GBF(EN,U)
       IMPLICIT REAL*8 (A-H,O-Z)
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 FUNCTION: GBF *************************
C
C PURPOSE Calculate bound-free gaunt factor as given by Eqn. 34 in
C         Burgess and Summers, MNRAS, vol 226, p257-272 (1987).
C
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 16-01-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST VERSION
C
C VERSION: 1.2                          DATE: 13-10-99
C MODIFIED: Martin O'Mullane
C               - Annotated routine
C               - Put in double precision constants.
C               - Corrected T2 correction factor for an erroneous 25/18
C                 to 28/18 as in the paper.
C-----------------------------------------------------------------------
C
       X=(EN*(U+1.0D0))**0.6666667D0
       T1=0.1728D0*(U-1.0D0)/X
       T2=1.55555556D0*T1*T1+1.333333D0*(0.0496D0*
     &    (U*U+1.333333D0*U+1.0D0))/(X*X)
       GBF=1.0D0/(1.0D0-1.333333D0*T1+T2)**0.75D0
       RETURN
       END
      REAL*8              EN,          U
```

4.117 gentab: Subroutine gentab from library adas3xx

```
       SUBROUTINE GENTAB(NREP,IMAX,DENS,TE,BMENER,F1,F2,F3)
C
       IMPLICIT REAL*8(A-H,O-Z)
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: GENTAB *********************
C
C PURPOSE : SEND DATA TO STREAM '10' FOR SUBSEQUENT TABLE PRODUCTION
C
C NOTES: THIS ROUTINE IS NOT YET PROPERLY ANNOTATED
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 25-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST VERSION
C
C-----------------------------------------------------------------------
C
      INTEGER             IMAX,        NREP(31)
      REAL*8              BMENER,      DENS,        F1(30),      F2(30)
      REAL*8              F3(30),      TE
```

4.118 ghnle: Subroutine ghnle from library adas3xx

```
       SUBROUTINE GHNLE(Z0,Z1,ZEFF,N,L,MULTN,N1,MULTN1,NL2,TEV,GAMA,
     *GAMTOT)
       IMPLICIT REAL*8(A-H,O-Z)
C-----------------------------------------------------------------------
C  PURPOSE: EVALUATES APPROXIMATE EXCITATION RATE PARAMETERS, GAMMA,
C  FROM N,L (SINGLET OR TRIPLET) LEVELS OF HELIUM-LIKE IONS TO
C  HIGHER LEVELS N1,L1 (SINGLET OR TRIPLET) USING CLASSICAL OVERLAPS.
C
C  FOR TRANSITIONS FROM 1S,2S AND 2P APPROXIMATE FITTINGS ARE USED BASED
C  ON SAMPSON DATA.(FOR SINGLET OR TRIPLET LEVELS)
C  FOR TRANSITIONS FROM HIGHER NL LEVELS UPWARDS A RESOLVED VARIANT
C  OF PERCIVAL-RICHARDS IS USED, WITH NUMERICAL QUADRATURES.
C  ALLOWENCE HAS BEEN MADE FOR THE EFFECT OF SPIN CHANGE.
C
C  *******   H.P. SUMMERS, JET   15 JAN. 1985  ******************
C  ******* SPENCE+SUMMERS(1985),  PAPER TO BE PUBLISHED *********
C  *******     J. SPENCE,  STRATHCLYDE    18 NOV. 1985  *********
C  INPUT
C      Z0=NUCLEAR CHARGE
C      Z1=ION CHARGE+1
C      ZEFF=EFFECTIVE ION CHARGE   (CF. SAMPSON ET AL.)
C      N=LOWER PRINCIPAL QUANTUM NUMBER
C      L=LOWER ANGULAR QUANTUM NUMBER
C      MULTN=MULTIPLICITY OF N
C      N1=UPPER PRINCIPAL QUANTUM NUMBER
C      MULTN1=MULTIPLICITY OF N1
C      NL2=INCLUSIVE LIMIT FOR THE RESOLUTION OF N INTO SEPERATE L'S.
C      TEV=ELECTRON TEMPERATURE (EV)
C  OUTPUT
C      GAMA(I),I=1,N1 IS VECTOR OF RATE PARAMETERS WITH L1=I-1 AND
C                        L1 THE UPPER ANGULAR QUANTUM NUMBER
C      GAMTOT = SUM OF GAMA(I),I=1,N1
C
C
C
C  VERSION : 1.2
C  MODIFIED: Martin O'Mullane
C  DATE    : 08-11-2004
C            Alter nmax in gamaf() from 200 to 500.
C
C  VERSION : 1.3
C  MODIFIED: Allan Whiteford
C  DATE    : 16-05-2007
C            - Updated comments as part of subroutine documentation
C              procedure.
C
C-----------------------------------------------------------------------
      INTEGER             L,           MULTN,       MULTN1,      N
      INTEGER             N1,          NL2
      REAL*8              GAMA(20),    GAMTOT,      TEV,         Z0
      REAL*8              Z1,          ZEFF
```

4.119 ghnlv: Subroutine ghnlv from library adas3xx

```
       SUBROUTINE GHNLV(Z0,Z1,ZEFF,N,L,N1,NL2,TEV,GAMA,GAMTOT)
C-----------------------------------------------------------------------
C  PURPOSE: EVALUATES APPROXIMATE EXCITATION RATE PARAMETERS, GAMMA,
C  FROM N,L LEVELS OF HYDROGEN-LIKE AND LITHIUM-LIKE IONS TO HIGHER
C  LEVELS N1,L1 USING CLASSICAL OVERLAPS.
C
C  FOR TRANSITIONS FROM 1S,2S AND 2P APPROXIMATE FITTINGS ARE USED BASED
C  ON SAMPSON DATA.
C  FOR TRANSITIONS FROM HIGHER NL LEVELS UPWARDS A RESOLVED VARIANT
C  OF PERCIVAL-RICHARDS IS USED, WITH NUMERICAL QUADRATURES.
C  *******   H.P. SUMMERS, JET   15 JAN. 1985  ******************
C  INPUT
C      Z0=NUCLEAR CHARGE
C      Z1=ION CHARGE+1
C      ZEFF=EFFECTIVE ION CHARGE   (CF. SAMPSON ET AL.)
C      N=LOWER PRINCIPAL QUANTUM NUMBER
C      L=LOWER ANGULAR QUANTUM NUMBER
C      N1=UPPER PRINCIPAL QUANTUM NUMBER
C      NL2=MAXIMUM INCLUSIVE N FOR WHICH N WILL BE RESOLVED
C      TEV=ELECTRON TEMPERATURE (EV)
C  OUTPUT
C      GAMA(I),I=1,N1 IS VECTOR OF RATE PARAMETERS WITH L1=I-1 AND
C                        L1 THE UPPER ANGULAR QUANTUM NUMBER
C      GAMTOT = SUM OF GAMA(I),I=1,N1
C-----------------------------------------------------------------------
C VERSION : 1.2
C MODIFIED: Martin O'Mullane
C       DATE    : 08-11-2004
C   Alter nmax in gamaf() from 200 to 500.
C
C VERSION  : 1.3
C DATE     : 16-05-2007
C MODIFIED : Allan Whiteford
C                 Updated comments as part of subroutine documentation
C                 procedure.
C-----------------------------------------------------------------------
       IMPLICIT REAL*8(A-H,O-Z)
      INTEGER             L,           N,           N1,          NL2
      REAL*8              GAMA(20),    GAMTOT,      TEV,         Z0
      REAL*8              Z1,          ZEFF
```

4.120 gspc: Subroutine gspc from library adas3xx

```
       SUBROUTINE GSPC(XA,N,C1,C2,C3,C4)
C
       IMPLICIT REAL*8(A-H,O-Z)
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: GSPC ***********************
C
C PURPOSE UNKNOWN
C
C NOTES: THIS ROUTINE IS NOT YET PROPERLY ANNOTATED
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 18-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - PUT UNDER SCCS CONTROL
C
C-----------------------------------------------------------------------
C
      INTEGER             N
      REAL*8              C1(10,9),    C2(10,9),    C3(10,9)
      REAL*8              C4(10,9),    XA(10)
```

4.121 hydemi: Subroutine hydemi from library adas3xx

```
      SUBROUTINE HYDEMI(lpass, iunt)

      IMPLICIT REAL*8 (A-H,O-Z)
C-----------------------------------------------------------------------
C     PURPOSE: Calculates the collisional mixing of excited levels of
C              hydrogen atoms in the beam.
C
C     IMAX - number of Stark resolved states
C     IMAXB - total number of included states
C     NA(I)- an array with principle quantum numbers of the states
C     ESTK(I)- the energies of the states in rydbergs
C     A(I,J) -|<r ij>|**2  - dipol matrix elements in at.units
C     BK(I) - the results of calculations
C     EBEAM - the energy of the beam atoms (keV/nucleon)
C     T - electron temperature of plasma in eV
C     DENS - electron density ( cm -3 )
C     ZEFF - effective charge of the plasma
C     MU - reduced mass(in mass of electron)
C-----------------------------------------------------------------------
C
C ADAS305 version. Developed from JETSHP.STARK.FORT (H P Summers).
C
C VERSION  : 1.1
C DATE     : 24-02-2005
C MODIFIED : Martin O'Mullane
C             - First version.
C
C VERSION  : 1.2
C DATE     : 16-03-2005
C MODIFIED : Martin O'Mullane
C             - Declare lpass as logical.
C
C VERSION  : 1.3
C DATE     : 16-05-2007
C MODIFIED : Allan Whiteford
C             - Updated comments as part of subroutine documentation
C               procedure.
C
C-----------------------------------------------------------------------
C----- alterations to dimensions and common to allow bundle-n extension
C----- hps 6 dec 1988
C-----------------------------------------------------------------------
       logical   lpass
C-----------------------------------------------------------------------
      INTEGER             IUNT
      LOGICAL             LPASS
```

4.122 initpos: Subroutine initpos from library adas3xx

```
      SUBROUTINE INITPOS(NTA,M)
C
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: INITPOS ********************
C
C PURPOSE UNKNOWN
C
C NOTES: THIS ROUTINE IS NOT YET PROPERLY ANNOTATED
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 01-02-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - PUT UNDER SCCS CONTROL
C
C-----------------------------------------------------------------------
C
      INTEGER             M,           NTA(*)
```

4.123 lftime: Subroutine lftime from library adas3xx

```
       SUBROUTINE LFTIME(ZEFF,N,L,NMIN,TAU)
       IMPLICIT REAL*8(A-H,O-Z)
C-----------------------------------------------------------------------
C
C  PURPOSE: EVALUATES RADIATIVE LIFETIME OF AN NL LEVEL OF A H-LIKE ION.
C
C  *************** H.P. SUMMERS, JET    22 JAN 1985  *****************
C  *** CORRECTIONS 13/5/85
C  *** CORRECTIONS 22/5/85
C  INPUT
C       ZEFF=EFFECTIVE CHARGE FOR OUTER ELECTRON
C       N=PRINCIPAL QUANTUM NUMBER FOR OUTER ELECTRON
C       L=ORBITAL ANGULAR MOMENTUM QUANTUM NUMBER FOR OUTER ELECTRON
C       NMIN=LOWEST ACCESSIBLE PRINCIPAL QUANTUM NUMBER BY DIPOLE
C            TRANSITION
C  OUTPUT
C       TAU=LIFETIME (SEC)
C
C VERSION  : 1.2
C DATE     : 16-05-2007
C MODIFIED : Allan Whiteford
C             - Updated comments as part of subroutine documentation
C               procedure.
C
C-----------------------------------------------------------------------
       IF(N-NMIN)5,5,10
    5  TAU=1.0D10
       RETURN
   10  NMAX=N-1
       XN=N
       XL=L
       T1=2.67744D9*ZEFF**4/(2.0D0*XL+1.0D0)
       SUM=0.0D0
       DO 20 N1=NMIN,NMAX
       XN1=N1
       DE=1.0D0/XN1**2-1.0D0/XN**2
       DE3=DE**3
       L1=L+1
       XL1=L1
       IF(L1.GE.N1)GO TO 15
       SUM=SUM+XL1*T1*DE3*RD2BS(N1,L1,N,L)
   15  L1=L-1
       XL1=L1
       IF(L1.LT.0.OR.L1.GE.N1)GO TO 20
       SUM=SUM+XL*T1*DE3*RD2BS(N1,L1,N,L)
   20  CONTINUE
       IF(SUM.LE.1.0D-10)GOTO 5
       TAU=1.0D0/SUM
       RETURN
      END
      INTEGER             L,           N,           NMIN
      REAL*8              TAU,         ZEFF
```

4.124 lowpop: Subroutine lowpop from library adas3xx

```
       SUBROUTINE LOWPOP(SEQ,REFMEM,Z0,Z1,ZEFF,NMET,IMETR,TE,DENS,TP,
     &                   DENSP,DENSH,NPARNT,DEPA,LPTA,
     &                   PEXMAT,PEXRHS,PECION,PEDREC,PERREC,
     &                   PEXREC,IECION,IEDREC,IERREC,IEXREC)
       IMPLICIT REAL*8(A-H,O-Z)
C -----------------------------------------------------------------------
C  PURPOSE: CALCULATE POPULATIONS OF LOW EXCITED POPULATIONS OF IONS
C
C  INCLUDING
C
C           (A) COUPLING TO HIGH LEVELS AND CONTINUUM VIA PRELIMINARY
C               BUNDLE-N CALCULATION (V2BNDLN) AND PROJECTION/EXPANSION
C               MATRIX CALCULATION (CLDLBN2)
C
C           (B) DEPENDENCE ON METASTABLES.
C
C
C  PROCESSES CAN INCLUDE ELECTRON & PROTON IMPACT, SPONTANEOUS EMISSION,
C  FREE ELECTRON RECOMBINATION AND CHARGE EXCHANGE RECOMBINATION
C  DEPENDING ON THE INPUT DATA SET
C
C  THE BASIC LOW LEVEL ATOMIC DATA IS ENTERED FROM COMPILATION
C  DATA SETS OF THE FORM
C
C             'JETSHP.<SE>LIKE.DATA(<TITLE><EL>)'
C
C       WHERE <SE> DENOTES THE ISOLELECTRONIC SEQUENCE, <TITLE> IS AN
C       ARBITRARY IDENTIFIER FOR THE SOURCE AND YEAR, <EL> IS THE
C       ELEMENT SYMBOL. <SE> AND <TITLE><EL> ARE PRESENTED TO THE
C       SUBROUTINE AS PARAMETERS FROM CLDLBN2
C
C
C  MAIN OUTPUT TABLES ARE GENERATED ON STREAM 7
C  POPULATION DATA FOR DIAGNOSTIC USE ON STREAM 19
C
C  THE SUBROUTINE WAS DEVELOPED FROM THE ORIGINAL PROGRAM SPMETPOP
C
C
C  INPUT
C      Z0      = NUCLEAR CHARGE
C      Z1      = RECOMBINING ION CHARGE
C      ZEFF    = PLASMA Z EFFECTIVE
C      NMET    = NUMBER OF METASTABLES (1.LE.NMET.LE.5)
C      IMETR(I)= INDEX OF METASTABLE I IN COMPLETE LEVEL LIST
C      TE      = ELECTRON TEMPERATURE (K)
C      DENS    = ELECTRON DENSITY (CM-3)
C      TP      = PROTON TEMPERATURE   (K)
C      DENSP   = PROTON DENSITY (CM-3)
C      DENSH   = NEUTRAL HYDROGEN DENSITY(IN BEAMS)  (CM-3)
C      PEXMAT(I,J) = EXPANDED PROJECTION MATRIX FOR LOW LEVELS
C      PEXRHS(I)   = EXPANDED PROJECTED RIGHT HAND SIDE VECTOR
C      PECION(I)   = DENS*IONISATION RATE COEFFT  FOR LOW LEVEL I
C      PEDREC(I)   = DIELECTRONIC RECOMB. COEFFT. FOR LOW LEVEL I
C      PERREC(I)   = RADIATIVE RECOMB. COEFFT.    FOR LOW LEVEL I
C      PEXREC(I)   = DENSH*(CX RATE COEFFT)/DENS  FOR LOW LEVEL I
C      IECION      = 0  IONISATION     ELIMINATED     FROM LEVEL I
C                  = 1  IONISATION     NOT ELIMINATED FROM LEVEL I
C      IEDREC      = 0  DIEL. RECOMB. ELIMINATED     FOR LEVEL I
C                  = 1  DIEL. RECOMB. NOT ELIMINATED FOR LEVEL I
```

```
C      IERREC      = 0  RAD.  RECOMB. ELIMINATED     FOR LEVEL I
C                  = 1  RAD.  RECOMB. NOT ELIMINATED FOR LEVEL I
C      IEXREC      = 0  CX.   RECOMB. ELIMINATED     FOR LEVEL I
C                  = 1  CX.   RECOMB. NOT ELIMINATED FOR LEVEL I
C
C
C
C  ********** H.P. SUMMERS, JET                26 APR 1990  **********
C         OPEN FILE ON STREAM 9 UNDER DIVUID   19 FEB 1991
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
C
C UPDATE:  19/01/94 - JONATHAN NASH - TESSELLA SUPPORT SERVICES PLC
C
C          THE FOLLOWING MODIFICATIONS HAVE BEEN MADE TO THE SUBROUTINE:
C
C          1) THE INPUT UNIT NUMBER WAS CHANGE FROM 10 TO 9 AND THE
C             DIAGNOSTIC OUTPUT UNIT FROM 11 TO 19.
C
C  NOTES:  NO ATTEMPT HAS BEEN MADE TO RESTRUCTURE THE ROUTINE. RATHER
C          THE MINIMUM AMOUNT OF WORK TO INTEGRATE THE ROUTINE INTO
C          ADAS310 HAS BEEN COMPLETED.
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 18-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - PUT UNDER SCCS CONTROL
C
C VERSION: 1.2                          DATE: 18-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - COMMENTED OUT LINE 'EXTERNAL GAMMA' AS IT APPEARS
C                 TO SERVE NO PURPOSE AND THERE IS NO CORRESPONDING
C                 GAMMA ROUTINE.
C
C VERSION: 1.3                          DATE: 18-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - COMMENTED OUT LINE 'CALL ERRSET'
C
C VERSION: 1.4                          DATE: 16-05-07
C MODIFIED: Allan Whiteford
C             - Updated comments as part of subroutine documentation
C               procedure.
C
C-----------------------------------------------------------------------
      CHARACTER*8         REFMEM
      CHARACTER*2         SEQ
      INTEGER             IECION,      IEDREC,      IERREC,      IEXREC
      INTEGER             IMETR(NDMET+1),           LPTA(NDLEV), NMET
      INTEGER             NPARNT
      REAL*8              DENS,        DENSH,       DENSP,       DEPA(5)
      REAL*8              PECION(NDLEV),            PEDREC(NDLEV)
      REAL*8              PERREC(NDLEV),            PEXMAT(NDLEV,NDLEV)
      REAL*8              PEXREC(NDLEV),            PEXRHS(NDLEV)
      REAL*8              TE,          TP,          Z0,          Z1
      REAL*8              ZEFF
```

4.125 lumsis: Subroutine lumsis from library adas3xx

```
      SUBROUTINE LUMSIS(N)

      IMPLICIT REAL*8 (A-H,O-Z)
C-----------------------------------------------------------------------
C PURPOSE: Finds the solutions of a set of linear equations
C
C N- a number of equations inthe set + 1
C AK(I,J)-coefficients in equations
C BK(I) - right side terms of equations
C When solutions are found they will be written into array BK(I) in
C the common block
C
C-----------------------------------------------------------------------
C
C ADAS305 version. Developed from JETSHP.STARK.FORT (H P Summers).
C
C VERSION  : 1.1
C DATE     : 24-02-2005
C MODIFIED : Martin O'Mullane
C             - First version.
C
C VERSION  : 1.2
C DATE     : 16-05-2007
C MODIFIED : Allan Whiteford
C             - Updated comments as part of subroutine documentation
C               procedure.
C
C-----------------------------------------------------------------------
      COMMON /CX/ AK(65,65),BK(65)
C*********
C**   EXPANSION OF MATRIX AK
C*********

      DO 200 I=1,N-1
      AK(N,I)=FLOAT(I)
200   AK(I,N)=BK(I)

C*****   MAX OF MAIN MINORS

      DO 202 K=1,N-2
      AMAX=AK(K,K)
      IMAX=K
      JMAX=K
      DO 206 I=K,N-1
      DO 206 J=K,N-1
      IF (DABS(AMAX).GE.DABS(AK(I,J))) GOTO 206
      AMAX=AK(I,J)
      JMAX=J
      IMAX=I
206   CONTINUE

C***** LINES INTERCHANGE

      IF(IMAX.EQ.K) GOTO 210
      DO 212 J1=1,N
      AB=AK(IMAX,J1)
      AK(IMAX,J1)=AK(K,J1)
212   AK(K,J1)=AB
```

```
C***** ROWS INTERCHANGE

210   IF(JMAX.EQ.K) GOTO 202
      DO 216 I1=1,N
      AB=AK(I1,JMAX)
      AK(I1,JMAX)=AK(I1,K)
216   AK(I1,K)=AB
202   CONTINUE

C******* FIRST FORMATION BK

      DO 218 I=1,N-1
218   BK(I)=AK(I,N)

C*****
C****   LU EXPANSION
C*********

      DO 120 K=1,N-2
      AB=1.E00/AK(K,K)
      DO 122 J=K,N-2
122   AK(K,J+1)=AK(K,J+1)*AB

      DO 120 J=K+1,N-1
      DO 120 I=K+1,N-1
120   AK(I,J)=AK(I,J)-AK(I,K)*AK(K,J)

C**** SOLUTION OF THE SYSTEM

      BK(1)=BK(1)/AK(1,1)
      DO 105 I=2,N-1
      DO 106 J=1,I-1
106   BK(I)=BK(I)-AK(I,J)*BK(J)
105   BK(I)=BK(I)/AK(I,I)

      DO 107 I=N-2,1,-1
      DO 107 J=I+1,N-1
107   BK(I)=BK(I)-AK(I,J)*BK(J)

C*******     SECOND FORMATION OF BK

      DO 240 J=1,N-1
      DO 241 IB=1,N-1
      IF(IB.NE.IDINT(DABS(AK(N,J)))) GOTO 241
      II=IB
      GOTO 240
241   CONTINUE
240   AK(II,N)=BK(J)
      DO 242 J=1,N-1
242   BK(J)=AK(J,N)
      RETURN
      END
      INTEGER             N
```

4.126 matinl: Subroutine matinl from library adas3xx

```
      SUBROUTINE MATINL(A,N,B,M,DETERM)
       IMPLICIT REAL*8(A-H,O-Z)
C-----------------------------------------------------------------------
C     PURPOSE: MATRIX INVERSION WITH ACCOMPANYING SOLUTION
C              OF LINEAR EQUATIONS
C
C VERSION  : 1.1
C DATE     : 18-03-1999
C MODIFIED : ???
C
C VERSION  : 1.2
C DATE     : 16-05-2007
C MODIFIED : Allan Whiteford
C             - Updated comments as part of subroutine documentation
C               procedure.
C
C-----------------------------------------------------------------------
      INTEGER             M,           N
      REAL*8              A(160,160),  B(160),      DETERM
```

4.127 nhydes: Subroutine nhydes from library adas3xx

```
       SUBROUTINE NHYDES(Z0,ZEFF,N,L,E0)
       IMPLICIT REAL*8(A-H,O-Z)
C-----------------------------------------------------------------------
C
C  PURPOSE: CALULATES LOWEST ORDER NON-RELATIVISTIC, RELATIVISTIC AND
C           QUANTUM-ELECTRODYNAMIC ENERGIES FOR HYDROGENIC IONS
C
C  BINDING ENERGY FOR CENTRE OF TERM IS PRODUCED.  FINE STRUCTURE FOR
C  L>0 MUST BE ADDED EXTERNALLY. FORMULAE ARE FROM ERIKSON (1977) J.PHY
C  CHEM. REF.DATA,6,831.
C  QED EFFECTS FOR L>0 OMITTED.
C-----------------------------------------------------------------------
C VERSION  : 1.1
C DATE     : 18-03-1999
C MODIFIED : ???
C
C VERSION  : 1.2
C DATE     : 05-10-2000
C MODIFIED : ???
C             - Removed junk from columns > 72
C
C VERSION  : 1.3
C DATE     : 16-05-2007
C MODIFIED : Allan Whiteford
C             - Updated comments as part of subroutine documentation
C               procedure.
C
C-----------------------------------------------------------------------
      INTEGER             L,           N
      REAL*8              E0,          Z0,          ZEFF
```

4.128 nlthes: Subroutine nlthes from library adas3xx

```
       SUBROUTINE NLTHES(Z0,ZEFF,N,L,E0)
       IMPLICIT REAL*8(A-H,O-Z)
C-----------------------------------------------------------------------
C  PURPOSE: PROVIDES BINDING ENERGY OF TERM CENTRE FOR OUTER ELECTRON
C           IN LITHIUM LIKE IONS
C
C  FROM EDLEN (1979) PHYSICA SCRIPTA, 19, 255.
C
C  FINE STRUCTURE FOR L>0 MUST BE ADDED EXTERNALLY. INFINITE MASS VALUE
C  FOR RYDBERG CONSTANT IS USED. (NOT NOW SEE BELOW!!)
C                                    ---
C  5 SEPT 1985 CHANGED R =109737.3 (I.E INFINITY MASS VALUE) TO
C  ----------- THE Z DEPENDANT EQUATION SEE BELOW.......J.SPENCE
C
C       ALL NEWLLPS3  RUNS  FROM  5 SEPT 1985 ONWARDS HAVE THIS
C       CHANGE ADDED IN, BUT %DIFF. IS SO VERY VERY SLIGHT THAT
C       EVEN IF  THE FINAL  RESULTS  DO  CHANGE THEY WILL BE IN
C       THE LAST FEW DEC. PLACES.
C         E.G BEFORE 3.11321270 NOW GET 3.11321269 ===> APPROX. N/C
C
C    NEWLLPS3.FORT(NLTHES2) ====> IS THE ORIGINAL ROUTINE WITH
C                                 RZ=109737.318.
C                                 (THIS ROUTINE EXISTS IN SOURCE FORM
C                                  ONLY. I.E IT IS NOT COMPILED.)
C
C-----------------------------------------------------------------------
C VERSION  : 1.1
C DATE     : 18-03-1999
C MODIFIED : ???
C
C VERSION  : 1.2
C DATE     : 05-10-2000
C MODIFIED : ???
C             - Removed junk from columns > 72
C
C VERSION  : 1.3
C DATE     : 16-05-2007
C MODIFIED : Allan Whiteford
C             - Updated comments as part of subroutine documentation
C               procedure.
C
C-----------------------------------------------------------------------
       EL0(Z)=0.25D0*(Z*Z-3.18244D0*Z+2.0038D0+0.208015D0/(Z-1.3833D0))
     &-6.3789D-6*RCORR*(Z-2.0D0)**2
       SIG0(Z)=0.141441D0*(Z-1.7025D0-0.768371D0*(1.0D0-0.090333D0/(Z-
     &2.184D0))/(Z-0.975D0))-1.9137D-6*RCORR*(Z-2.0D0)**2
       P(XN,XL)=(3.0D0*XN*XN-XL*(XL+1.0D0))/(2.0D0*XN**5*(XL-0.5D0)*
     &XL*(XL+0.5D0)*(XL+1.0D0)*(XL+1.5D0))
       QP(XN,XL)=(35.0D0*XN**4-5.0D0*XN*XN*(6.0D0*XL*(XL+1.0D0)-5.0D0)+
     &3.0D0*(XL-1.0D0)*XL*(XL+1.0D0)*(XL+2.0D0))/(8.0D0*XN**7*(XL-1.5D0)
     &*(XL-1.0D0)*(XL-0.5D0)*XL*(XL+0.5D0)*(XL+1.0D0)*(XL+1.5D0)*(XL+
     &2.0D0)*(XL+2.5D0))
       DRS(Z)=0.456536D0*(Z-1.2808D0)**4+1.2763D-5*(Z-1.2808D0)**6+
     &4.34D-10*(Z-1.2808D0)**8
       DRP(Z)=0.21305D0*(Z-2.241D0)**4+0.466D-5*(Z-2.241D0)**6+
     &1.48D-10*(Z-2.241D0)**8
       DLS(Z)=4.5246D-3*(Z-1.6D0)**4*(-2.179D0-2.0D0*DLOG(7.29729D-3*
     &(Z-1.6D0))+5.26427D-2*(Z-1.6D0)-5.32504D-5*(Z-1.6D0)**2*(3.0D0*
     &(DLOG(7.29729D-3*(Z-1.6D0)))**2+8.695D0*DLOG(7.29729D-3*(Z-1.6D0))
```

```
     &+19.081D0))
       DLP(Z)=4.525D-3*(Z-2.0D0)**4*(3.0D-2-2.6412D-5*(Z-2.0D0)**2*DLOG(
     &7.29729D-3*(Z-2.0D0)))
C
C           ................. IMPROVEMENTS ...................
C
C      R=109737.318D0  <<<<<<< OLD LINE BEFORE 5 SEPT 1985.
C      REPLACEMENT FOR ABOVE LINE IS GIVEN BELOW.
       R=109737.318D0-60.200D0/(2.00D0*Z0)
C
C      EL0( ) AND SIG0( ) ABOVE HAD SOME CONSTANTS DIVIDED BY 109737.3
C      THIS CORRECTION VALUE PUTS 109737.3 BACK IN AND INSTEAD DIVIDES
C      THOSE CONSTANTS BY THE Z-DEPT R ABOVE.
       RCORR=109737.3/R
C
C      WRITE(6,1414)R,RCORR
 1414  FORMAT(' R,RCORR = ',1P2E15.7)
       EN=N
       XL=L
       T1=DRS(Z0)
       T2=DRP(Z0)
       T3=DLS(Z0)
       T4=DLP(Z0)
C      WRITE(6,101)T1,T2,T3,T4
       IF(L-1)10,20,40
C  S STATES
   10  T2S=R*EL0(Z0)+DRS(Z0)-DLS(Z0)
C      WRITE(6,101)T2S
       T2S=T2S/(R*(Z0-2.0D0)**2)
C      WRITE(6,101)T2S
       D2S=2.0D0-1.0D0/DSQRT(T2S)
       C=0.0828D0/Z0-0.2283D0/Z0**2
       B=-5.5D-4*Z0+5.963D-3+0.19404D0/(Z0-0.36D0)-0.3368D0/(Z0-0.36D0)*
     &*2
       A=D2S-B*T2S-C*T2S*T2S
C      WRITE(6,101)A,B,C
       GO TO 30
C  P STATES
   20  T2P1=R*EL0(Z0)+DRS(Z0)-DLS(Z0)
       T2P2=R*SIG0(Z0)+DRS(Z0)-DRP(Z0)-DLS(Z0)+DLP(Z0)
C      WRITE(6,101)T2P1,T2P2
       T2P=(T2P1-T2P2)/(R*(Z0-2.0D0)**2)
       D2P=2.0D0-1.0D0/DSQRT(T2P)
       C=-2.603D-2/(Z0-2.37D0)+1.326D-2/(Z0-2.37D0)**2
       B=-1.2D-3*Z0+2.1237D-2-8.905D-2/(Z0-1.74D0)+4.803D-2/(Z0-1.74D0)*
     &*2
       A=D2P-B*T2P-C*T2P*T2P
C      WRITE(6,101)A,B,C
   30  U=0.0D0
   31  U0=U
       V=EN-U0
       TNL=1.0D0/V**2
       U=A+TNL*(B+TNL*C)
C      WRITE(6,101)TNL,U,U0
  101  FORMAT(1P4D15.7)
       IF(DABS(U-U0).LE.1.0D-6)GO TO 35
       GO TO 31
   35  TNL=((Z0-2.0D0)/(EN-U))**2
       GO TO 50
C  L>2 CASES
   40  S=0.3397D0+0.102D0/(Z0-0.4D0)
```

```
       A=9.0D0*((Z0-2.0D0)/(Z0-S))**4
       AK=0.2113D0*Z0+0.598D0-2.4D0/Z0
       DELTAP=A*P(EN,XL)+A*DSQRT(A)*AK*QP(EN,XL)
C      TRR=(Z0-2.0D0)**2*(1.0D0+5.32504D-5*(Z0-2.0D0)**2*(EN/(XL+0.5D0)-
C    &0.75D0)/(EN*EN))/(EN*EN)
C      WRITE(6,7845)TRR*R,DELTAP*R
 7845  FORMAT(' TRR*R,DELTAP*R = ',1P2E15.7)
       TNL=(Z0-2.0D0)**2*(1.0D0+5.32504D-5*(Z0-2.0D0)**2*(EN/(XL+0.5D0)-
     &0.75D0)/(EN*EN))/(EN*EN)+DELTAP
   50  E0=TNL
C      WRITE(7,100)Z0,N,L,TNL
  100  FORMAT(1H ,F5.1,2I5,1PD15.7)
C      WRITE(6,7842)V
 7842  FORMAT(' N STAR = ',1PE15.7)
C      IF(L.LT.2)GOTO 7844
C      WRITE(6,7843)EN,XL,R*P(EN,XL),QP(EN,XL)/P(EN,XL)
 7843  FORMAT(' EN,XL,P,Q = ',1P4E15.7)
 7844  RETURN
      END
      INTEGER             L,           N
      REAL*8              E0,          Z0,          ZEFF
```

4.129 nsuph1: Subroutine nsuph1 from library adas3xx

```
       SUBROUTINE NSUPH1(TEV,EBEAM,TIEV,NIMP ,ZIMPA ,FRIMPA,AMIMPA,
     &                   ITYP1 ,ITYP2 ,ITYP3 ,ITYP4 ,ITYP5 ,ITYP6 ,
     &                   XTBE  ,XTBP  ,XTBZ  ,STBE  ,STBP  ,STBZ  ,
     &                   LXTBE ,LXTBP ,LXTBZ ,LSTBE ,LSTBP ,LSTBZ ,
     &                   PXTBE ,PXTBP ,PXTBZ ,PSTBE ,PSTBP ,PSTBZ ,
     &                   LPXTBE,LPXTBP,LPXTBZ,LPSTBE,LPSTBP,LPSTBZ,
     &                   DSLPATH)
C
       IMPLICIT REAL*8(A-H,O-Z)
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: NSUPH1 *********************
C-----------------------------------------------------------------------
C  PURPOSE: ACCESS SPECIFIC HIGHER QUALITY DATA FOR HYDROGEN
C
C  POPULATION STRUCTURE CALCULATION IN THE BUNDLE-N APPROXIMATION.
C
C  DATA TYPES ARE:
C
C         (1) ELECTRON IMPACT EXCITATION - SPECIFIC ION FILE IS OPENED.
C         (2) ELECTRON IMPACT IONISATION - SPECIFIC FIT IS USED.
C         (3) H+  IMPACT EXCITATION      - QHIEXDAT FILE IS OPENED.
C         (4) H+  IMPACT IONIS + CX      - QHIEXDAT FILE IS OPENED.
C         (5) ZIMP ION IMPACT EXCITATION - QHIEXDAT FILE IS OPENED.
C         (6) ZIMP ION IMPACT IONIS + CX - QHIEXDAT FILE IS OPENED.
C
C  INPUT
C      TEV      = ELECTRON TEMPERATURE (EV)
C      EBEAM    = BEAM ENERGY (EV/AMU)  USED AS A UNIFORM VELOCITY SHIFT
C                 FOR ION COLLISIONS
C      TIEV     = ION TEMPERATURE (EV)
C      NIMP     = NUMBER OF IMPURITY IONS (EXCLUDING H+)
C      ZIMPA()  = Z OF EFFECTIVE IMPURITY FOR ION COLLISIONS(EXCEPT H+)
C      FRIMPA() = FRACTION OF TOTAL IMPURITY NUMBER DENSITY (EXCL H+)
C      AMIMPA() = ATOMIC MASS NUMBER OF IMPURITY
C      ITYP1    = 0   DO NOT OBTAIN TYPE 1 DATA
C               = 1   OBTAIN TYPE 1 DATA
C      ITYP2    = 0   DO NOT OBTAIN TYPE 2 DATA
C               = 1   OBTAIN TYPE 2 DATA
C      ITYP3    = 0   DO NOT OBTAIN TYPE 3 DATA
C               = 1   OBTAIN TYPE 3 DATA
C      ITYP4    = 0   DO NOT OBTAIN TYPE 4 DATA
C               = 1   OBTAIN TYPE 1 DATA
C      ITYP5    = 0   DO NOT OBTAIN TYPE 5 DATA
C               = 1   OBTAIN TYPE 2 DATA
C      ITYP6    = 0   DO NOT OBTAIN TYPE 6 DATA
C               = 1   OBTAIN TYPE 3 DATA
C      DSLPATH  =     STRING CONTAINING PATH FOR INPUT FILE FOR UNIT 15
C
C  OUTPUT
C      XTBE(N,N'')  = TYPE 1 RATE COEFFICIENT
C      XTBP(N,N'')  = TYPE 3 RATE COEFFICIENT
C      XTBZ(N,N'')  = TYPE 5 RATE COEFFICIENT
C      STBE(N)      = TYPE 2 RATE COEFFICIENT
C      STBP(N)      = TYPE 4 RATE COEFFICIENT
C      STBZ(N)      = TYPE 6 RATE COEFFICIENT
C      LXTBE(N,N'') = TYPE 1 MARKER  (0 =NO VALUE, 1=VALUE)
C      LXTBP(N,N'') = TYPE 3 MARKER
```

```
C      LXTBZ(N,N'') = TYPE 5 MARKER
C      LSTBE(N)     = TYPE 2 MARKER
C      LSTBP(N)     = TYPE 4 MARKER
C      LSTBZ(N)     = TYPE 6 MARKER
C      PXTBE(N)     = TYPE 1 PROJECTION MULTIPLIER
C      PXTBP(N)     = TYPE 3 PROJECTION MULTIPLIER
C      PXTBZ(N)     = TYPE 5 PROJECTION MULTIPLIER
C      PSTBE        = TYPE 2 PROJECTION MULTIPLIER
C      PSTBP        = TYPE 4 PROJECTION MULTIPLIER
C      PSTBZ        = TYPE 6 PROJECTION MULTIPLIER
C      LPXTBE(N)    = TYPE 1 PROJECTION MULTIPLIER USED ABOVE THIS N'
C      LPXTBP(N)    = TYPE 3 PROJECTION MULTIPLIER USED ABOVE THIS N'
C      LPXTBZ(N)    = TYPE 5 PROJECTION MULTIPLIER USED ABOVE THIS N'
C      LPSTBE       = TYPE 2 PROJECTION MULTIPLIER USED ABOBE THIS N
C      LPSTBP       = TYPE 4 PROJECTION MULTIPLIER USED ABOVE THIS N
C      LPSTBZ       = TYPE 6 PROJECTION MULTIPLIER USED ABOBE THIS N
C
C  ********** H.P. SUMMERS, JET                 9 MAY 1990   **********
C  **********                                  20 JUL 1990   **********
C  **********                                  13 AUG 1990   **********
C  **********      NEW ELECTRON EXCIT. DATA     22 JAN 1991   **********
C  **********      NEW ION IMPACT EXCIT. DATA    3 JUL 1991   **********
C  **********      NEW ELEC. IMPACT ION. DATA    3 JUL 1991   **********
C  **********      DATA EXTENSION BY ADDING      1 MAR 1992   **********
C                  SOME INTERMEDIATE VALUES +
C                  ADDITION OF B, N, NE ION. +
C                  CHARGE EXCHANGE.
C  **********      MULTIPLE, SIMULTANEOUS       11 JAN 1994   **********
C                  IMPURITY EXTENSION
C                  ERROR CORRECTED IN IMPURITY
C                  REDUCED MASSES
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
C
C UPDATE:  19/01/94 - JONATHAN NASH - TESSELLA SUPPORT SERVICES PLC
C
C          THE FOLLOWING MODIFICATIONS HAVE BEEN MADE TO THE SUBROUTINE:
C
C          1) A PARAMETER FLAG HAS BEEN ADDED TO SWITCH ON/OFF
C             DIAGNOSTIC PRINTING (UNIT 6).
C
C  NOTES:  NO ATTEMPT HAS BEEN MADE TO RESTRUCTURE THE ROUTINE. RATHER
C          THE MINIMUM AMOUNT OF WORK TO INTEGRATE THE ROUTINE INTO
C          ADAS310 HAS BEEN COMPLETED.
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 16-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST VERSION
C
C VERSION: 1.2                          DATE: 18-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - ADDED VARIABLE DSLPATH AND CHANGED NAME OF INPUT FILE
C
C VERSION: 1.3                          DATE: 18-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - CORRECTED STRING HANDLING SYNTAX IN CONSTRUCTION OF
C                 DSNAME, COMMENTED OUT REFERENCES TO DEBUG LOGICAL
C                 VARIABLE AND INSERTED 'CALL' BEFORE XXSLEN.
```

```
C
C VERSION: 1.4                          DATE: 18-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - MODIFIED CONSTRUCTION OF DSNAME
C
C VERSION: 1.5                          DATE: 18-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - ADDED DSLPATH IN CALL TO QH.FOR
C
C VERSION: 1.6                          DATE: 22-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - REPLACED CALLS TO NAG ROUTINE E02BBF WITH ADAS ROUTINE
C                 DXNBBF
C
C VERSION: 1.7                          DATE: 23-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - REPLACED CALLS TO NAG ROUTINE E01BAF WITH ADAS ROUTINE
C                 DXNBAF
C
C VERSION: 1.8                          DATE: 08-02-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - REMOVED SUPERFLUOUS VARIABLES
C
C VERSION: 1.9                          DATE: 03-04-97
C MODIFIED: H.ANDERSON
C - ALTERED TO USE RESTRUCTURED ADF02 DATASET sia#h_rfm.dat
C
C VERSION: 1.10 DATE: 03/04/97
C MODIFIED: HARVEY ANDERSON.
C     ALTERED TO USE NEW PREFERRED ADF02 DATASET sia#h_j97.dat
C
C VERSION: 1.11 DATE: 08-04-97
C MODIFIED: RICHARD MARTIN
C     CHANGED NAME OF ADF02 FILE FROM sia#h_j97.dat TO
C     sia#h_j97#h.dat
C
C VERSION: 1.12 DATE: 23-02-99
C MODIFIED: HARVEY ANDERSON
C     ADDED ADDITIONAL CODE TO ACCESS THE FUNDAMENTAL DATA
C     FOR ARGON WHICH IS CONTAINED IN THE ADF02 TYPE FILE.
C
C
C VERSION : 1.13                        DATE: 20-10-2003
C MODIFIED: Martin O'Mullane
C           - Extend TITLX to 120 to match e2titl routine.
C
C VERSION: 1.14 DATE: 07-07-2004
C MODIFIED: Allan Whiteford
C           - Changed calls from DXNB{A,B}F TO XXNB{A,B}F
C
C VERSION: 1.15 DATE: 07-07-2004
C MODIFIED: Allan Whiteford
C           - Updated comments as part of subroutine documentation
C             procedure.
C
C-----------------------------------------------------------------------
C
C  PARAM : (L*4)  DEBUG     = FLAGS DIAGNOSTIC PRINTING.
C                             .TRUE.  => PRINT DIAGNOSTICS.
C                             .FALSE. => DO NOT PRINT DIAGNOSTICS.
C
```

```
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      LOGICAL    DEBUG
      CHARACTER*80        DSLPATH
      INTEGER             ITYP1,       ITYP2,       ITYP3,       ITYP4
      INTEGER             ITYP5,       ITYP6,       LPSTBE,      LPSTBP
      INTEGER             LPSTBZ,      LPXTBE(NDLOW)
      INTEGER             LPXTBP(NDLOW),            LPXTBZ(NDLOW)
      INTEGER             LSTBE(NDLOW),             LSTBP(NDLOW)
      INTEGER             LSTBZ(NDLOW),             LXTBE(NDLOW,NDLOW)
      INTEGER             LXTBP(NDLOW,NDLOW),       LXTBZ(NDLOW,NDLOW)
      INTEGER             NIMP
      REAL*8              AMIMPA(10),  EBEAM,       FRIMPA(10),  PSTBE
      REAL*8              PSTBP,       PSTBZ,       PXTBE(NDLOW)
      REAL*8              PXTBP(NDLOW),             PXTBZ(NDLOW)
      REAL*8              STBE(NDLOW), STBP(NDLOW), STBZ(NDLOW), TEV
      REAL*8              TIEV,        XTBE(NDLOW,NDLOW)
      REAL*8              XTBP(NDLOW,NDLOW),        XTBZ(NDLOW,NDLOW)
      REAL*8              ZIMPA(10)
```

4.130 nwripv: Subroutine nwripv from library adas3xx

```
       SUBROUTINE NWRIPV(IZ,WI,EI,WJ,EJ,M,PHI,EPS,OMEG,N,TVA,EM,IZC,RAT,
     &QI,QJ,GA)
C-----------------------------------------------------------------------
C   PURPOSE: CALCULATES ELECTRON & POSITIVE ION COLL. EXCITATION AND
C            DEEXCITATION RATE COEFFICIENTS FOR DIPOLE TRANSITIONS
C            IN THE IMPACT PARAMETER APPROXIMATION
C
C   (BURGESS AND SUMMERS,1976,MON.NOT.R.AST.SOC.,174,345)
C
C   OPTIONALLY A SET OF INCIDENT PARTICLE ENERGIES AND COLLISION
C   STRENGTHS MAY BE PROVIDED, IN WHICH CASE THE IMPACT PARAMETER THEORY
C   IS USED TO CALCULATE THE COLLISION STRENGTHS AT HIGH ENERGY
C   WITH VALUES SCALED TO THE HIGHEST ENERGY INPUT COLLISION STRENGTH.
C  ***************** H.P. SUMMERS, JET   22 JAN 1985  ******************
C
C  ****************   ORIGINAL VERSION WITHOUT THE   ******************
C  ****************  STATISTICAL WEIGHT (WT) CHANGE  ******************
C
C  INPUT
C       IZ=ION CHARGE
C       WI=STATISTICAL WEIGHT OF STATE I
C       EI=BINDING ENERGY OF STATE I (RYD)
C       WJ=STATISTICAL WEIGHT OF STATE J
C       EJ=BINDING ENERGY OF STATE J (RYD)
C       M=NUMBER OF TABULAR VALUES OF COLLISION STRENGTH
C       PHI=FIJ/EIJ WITH FIJ ABSORPTION OSCILLATOR STRENGTH
C                         EIJ=EI-EJ THE TRANSITION ENERGY (RYD)
C       EPS(K)=INCIDENT ELECTRON ENERGIES (RYDBERGS)
C       OMEG(K)=COLLISION STRENGTHS
C       N=NUMBER OF TEMPERATURES
C       TVA(I)=TEMPERATURES (EV) (INCIDENT PARTICLE DISTRIBUTION)
C       EM=REDUCED MASS FOR COLLIDING PARTICLE (ELECTRON MASSES)
C       IZC=CHARGE OF COLLIDING PARTICLE
C  OUTPUT
C       RAT=RATIO OF OMEG(M) TO I.P. OMEGA.
C       QI(I)=COLLISIONAL EXCITATION RATE COEFFICIENTS  (CM**3 SEC-1)
C       QJ(I)=COLLISIONAL DEEXCITATION RATE COEFFICIENTS.
C       GA(I)=GAMMA RATE PARAMETERS
C-----------------------------------------------------------------------
C  AUTHOR
C   HUGH SUMMERS     1977/5/20
C  UPDATES
C       1983/9/1, 1984/6/25
C  COMMENTS
C   I IS THE LOWER LEVEL OF THE TRANSITION.
C   M MAY BE ZERO, IN WHICH CASE NO EPS AND OMEG VALUES ARE REQUIRED.
C   UNDERFLOW IS NOT TRAPPED. THIS MAY BE ACHEIVED IN IBM FORTRAN WITH T
C   MODIFIED IPRATE TO ALLOW PROTON RATES. EM TAKEN AS INPUT AND
C   PHI ACCEPTED INPLACE OF AJI ON INPUT.
C-----------------------------------------------------------------------
C VERSION  : 1.1
C DATE     : 18-03-1999
C MODIFIED : ???
C
C VERSION  : 1.2
C DATE     : 05-10-2000
C MODIFIED : ???
C             - Removed junk from columns > 72
C
```

```
C VERSION  : 1.3
C DATE     : 16-05-2007
C MODIFIED : Allan Whiteford
C             - Updated comments as part of subroutine documentation
C               procedure.
C
C-----------------------------------------------------------------------
       IMPLICIT REAL*8(A-H,O-Z)
      INTEGER             IZ,          IZC,         M,           N
      REAL*8              EI,          EJ,          EM,          EPS(20)
      REAL*8              GA(40),      OMEG(20),    PHI,         QI(40)
      REAL*8              QJ(40),      RAT,         TVA(40),     WI
      REAL*8              WJ
```

4.131 omgrc2: Subroutine omgrc2 from library adas3xx

```
       SUBROUTINE OMGRC2(EI,EJ,WI,WJ,X,OMEG,ISW,M,ATE,QI,QJ)
       IMPLICIT REAL*8(A-H,O-Z)
C-----------------------------------------------------------------------
C  PURPOSE: GIVEN A SET OF VALUES OF COLLISION STRENGTHS AND INCIDENT
C           ELECTRON ENERGIES(IN THRESHOLD UNITS), THE ROUTINE
C           EVALUATES THE COLLISIONAL EXCITATION AND DEEXCITATION RATE
C           COEFFICIENTS.
C
C   EXTRAPOLATION PROCEDURE MAY BE SELECTED TO BE CONSTANT (ISW=1),
C   LOGARITHMIC (ISW=2) OR EXPONENTIAL (ISW=3).
C
C-----------------------------------------------------------------------
C VERSION  : 1.1
C DATE     : 18-03-1999
C MODIFIED : ???
C
C VERSION  : 1.2
C DATE     : 05-10-2000
C MODIFIED : ???
C             - Removed junk from columns > 72
C
C VERSION  : 1.3
C DATE     : 16-05-2007
C MODIFIED : Allan Whiteford
C             - Updated comments as part of subroutine documentation
C               procedure.
C
C-----------------------------------------------------------------------
      INTEGER             ISW,         M
      REAL*8              ATE,         EI,          EJ
      REAL*8              OMEG(20),    QI,          QJ,          WI
      REAL*8              WJ,          X(20)
```

4.132 outtmp: Subroutine outtmp from library adas3xx

```
      SUBROUTINE OUTTMP(TITLE,MN,RX3,DPT,EHCT,NHCT,LHCT,NIP,NEX,IPRT,
     &  NDEL,INTD,IPRS,ILOW,IONIP,NIONIP,ILPRS,IVDISP,ZEFF,
     &  DEDEG,NL1,NL2,NL3,Z0,Z1,ALF,AMSZ0,AMSHYD,LP,ISP,LMAX,
     &  NTAIL,LSWCH,IGHNL,IRNDGV,MJ,IREF,JREF,SREF,WREF,CREF,
     &  WIREF,WCREF,MG,IRA,TE,TP,TS,DENS,DENSP,W,BMENER,DENSH,
     &  NIMP,ZIMPA,AMIMPA,FRIMPA,ZP,EMP,NGA,LGA,ISGA,NGLA,INAL,
     &  INALTG,ANAEGY,ANAFPG,NSERA,LAA,LTAA,INBL,INBLT,INBIN0,
     &  INBN1,ANBE1,INBN2,ANBE2,INBN3,ANBE3,INREP,ILREP,ILTREP,
     &  NREP,INLREP1,INLMP,IUTMP,IDIEL,EIJA,FIJA,CORA,EIJSA,
     &  FIJSA,CORSA )
C-----------------------------------------------------------------------
C
C  ******************** FORTRAN 77 ROUTINE : OUTTMP *********************
C
C  PURPOSE : ROUTINE TO WRITE INPUT TO TEMPORARY FILE.
C
C
C  CONTACT : HARVEY ANDERSON
C            UNIVERSITY OF STRATHCLYDE
C            ANDERSON@BARWANI.PHYS.STRATH.AC.UK
C
C  DATE    : 04/21/98
C
C
C  VERSION : 1.2
C  MODIFIED: RICHARD MARTIN
C  DATE    : 21-03-2000
C            DECLARED LOGICAL VARIABLE LOPEN.
C
C  VERSION : 1.3
C  MODIFIED: Martin O'Mullane
C  DATE    : 3-6-2000
C            Tidied code to be within column 72.
C            TITLE was passed as c*6 but here was c*1 title(10).
C            Change to make them compatible and alter writing of
C            TITLE to file.
C
C  VERSION : 1.4
C  MODIFIED: Martin O'Mullane
C  DATE    : 20-01-2004
C            Further code tidying.
C
C-----------------------------------------------------------------------

      CHARACTER*6         TITLE
      INTEGER             IDIEL,       IGHNL,       ILOW,        ILPRS
      INTEGER             ILREP,       ILTREP,      INAL(2,10)
      INTEGER             INALTG(2,10),             INBIN0(2,10)
      INTEGER             INBL(2,10),  INBLT(2,10), INBN1(2,10)
      INTEGER             INBN2(2,10), INBN3(2,10), INLMP(10,10)
      INTEGER             INLREP1(10), INREP,       INTD,        IONIP
      INTEGER             IPRS,        IPRT,        IRA(20)
      INTEGER             IREF(10),    IRNDGV,      ISGA(2),     ISP
      INTEGER             IUTMP,       IVDISP,      JREF(10),    LAA(2)
      INTEGER             LGA(2),      LHCT,        LMAX,        LP
      INTEGER             LSWCH,       LTAA(2),     MG,          MJ
      INTEGER             MN,          NDEL,        NEX,         NGA(2)
      INTEGER             NGLA(2),     NHCT,        NIMP,        NIONIP
      INTEGER             NIP,         NL1,         NL2,         NL3
```

```
INTEGER            NREP(31),    NSERA(2),    NTAIL
REAL*8             ALF,         AMIMPA(10),  AMSHYD,      AMSZ0
REAL*8             ANAEGY(2,10),             ANAFPG(2,10)
REAL*8             ANBE1(2,10), ANBE2(2,10), ANBE3(2,10), BMENER
REAL*8             CORA(10,6),  CORSA(10,6), CREF(10),    DEDEG
REAL*8             DENS,        DENSH,       DENSP,       DPT
REAL*8             EHCT,        EIJA(10),    EIJSA(10),   EMP
REAL*8             FIJA(10),    FIJSA(10),   FRIMPA(10),  RX3
REAL*8             SREF(10),    TE,          TP,          TS
REAL*8             W,           WCREF(10),   WIREF(10)
REAL*8             WREF(10),    Z0,          Z1,          ZEFF
REAL*8             ZIMPA(10),   ZP
```

4.133 photo: Subroutine photo from library adas3xx

```
      SUBROUTINE PHOTO(PION,PREC,PSTIM,Z,TE,TP,EN,KPION,KPREC,KPSTIM)
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: PHOTO **********************
C
C PURPOSE UNKNOWN
C
C NOTES: THIS ROUTINE IS NOT YET PROPERLY ANNOTATED
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 16-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST VERSION
C
C-----------------------------------------------------------------------
C
      INTEGER             KPION,       KPREC,       KPSTIM
      REAL*8              EN,          PION,        PREC,        PSTIM
      REAL*8              TE,          TP,          Z
```

4.134 photo2: Subroutine photo2 from library adas3xx

```
      SUBROUTINE PHOTO2(PION,PREC,PSTIM,CREC,ZZ,TE,TP,N1,L1,LT1,E1,V1,
     1KPION,KPREC,KPSTIM,CUT,IOPT,C1,C2,C3)
C-----------------------------------------------------------------------
C
C ********** FORTRAN 77 ROUTINE : PHOTO2 **********
C
C PURPOSE : EVALUATES PHOTO IONISATION, STIMULATED
C   AND RADIATIVE RECOMBINATION RATES.
C
C NOTE    : THE INTEGRALS INVOLVING THE BOUND-FREE
C   GAUNT FACTOR ARE EVALUATED USING A
C   COMBINATION OF LAGUERRE AND GUASSIAN
C   QUADRATURE.
C
C HISTORY : ROUTINE WAS ORIGINALLY WRITTEN BY
C   H.P.SUMMERS.
C
C CONTACT : HARVEY ANDERSON
C   UNIVERSITY OF STRATHCLYDE
C   ANDERSON@PHYS.STRATH.AC.UK
C
C DATE    : 5/2/98
C
C-----------------------------------------------------------------------
C
C   CHANGE TO PHOTO2 ON 17/11/83 AND GPRT=GPRT1=0
       IMPLICIT REAL*8(A-H,O-Z)
      INTEGER             IOPT,        KPION,       KPREC,       KPSTIM
      INTEGER             L1,          LT1,         N1
      REAL*8              C1(5,5,3),   C2(5,5,3),   C3(5,5,3),   CREC
      REAL*8              CUT,         E1,          PION,        PREC
      REAL*8              PSTIM,       TE,          TP,          V1
      REAL*8              ZZ
```

4.135 pyip: Subroutine pyip from library adas3xx

```
       SUBROUTINE PYIP(D,EM,ZCOL,PHI,CPY,WI,WJ,R,TE,INTD,PY)
C
       IMPLICIT REAL*8 (A-H,O-Z)
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: PYIP ***********************
C
C PURPOSE: UNKNOWN
C
C NOTES: THIS ROUTINE IS NOT YET PROPERLY ANNOTATED
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 16-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST VERSION
C
C-----------------------------------------------------------------------
C
      INTEGER             INTD
      REAL*8              CPY,          D,            EM,           PHI
      REAL*8              PY,           R,            TE,           WI
      REAL*8              WJ,           ZCOL
```

4.136 pypr: Subroutine pypr from library adas3xx

```
       SUBROUTINE PYPR(E,E11,N,N11,EM,Z1,PHI,WI,WJ,TE,INTD,PY,RDEXC)
C
       IMPLICIT REAL*8(A-H,O-Z)
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: PYPR ***********************
C
C  PURPOSE: CALCULATES PY FACTOR (CF. VAN REGEMORTER,1962) USING
C           PERCIVAL,RICHARD AND COWORKER CROSS-SECTIONS.
C
C  VALID ONLY FOR ELECTRON INDUCED TRANSITIONS BETWEEN WHOLE PRINCIPAL
C  QUANTUM SHELLS IN HYDROGEN AND HYDROGENIC IONS, FOR N,N11>4
C  HOWEVER ADJUSTMENTS MADE TO ALLOW USE OF FORMULAE FOR N<4
C  ********* H.P.SUMMERS, JET  12 NOVEMBER 1984  ***************
C  INPUT
C      E=1/V**2 WITH V THE INITIAL EFFECTIVE PRINCIPAL QUANTUM NUMBER
C      E11=1/V11**2 WITH V11 THE FINAL EFFECTIVE PRINCIPAL QUANTUM NUMBE
C      N=INITIAL PRINCIPAL QUANTUM NUMBER
C      N11=FINAL PRINCIPAL QUANTUM NUMBER (REQUIRE N11>N AND V11>V)
C      EM=REDUCED MASS OF COLLIDING PARTICLE (MUST BE 1.0 IN THIS CASE)
C      Z1=TARGET ION CHARGE +1
C      PHI=(IH/EIJ)F WITH EIJ=TRANSITION ENERGY, F=ABS. OSCILL. STRENGTH
C      WI=STATISTICAL WEIGHT OF INITIAL LEVEL
C      WJ=STATISTICAL WEIGHT OF FINAL LEVEL
C      TE=ELECTRON TEMPERATURE(K)
C      INTD=<3 FOR TWO POINT GAUSSIAN QUADRATURE.
C           = 3 FOR THREE POINT GAUSSIAN QUADRATURE
C           =>3 FOR FOUR POINT GAUSSIAN QUADRATURE
C  OUTPUT
C      PY=P FACTOR
C      RDEXC=DEXCITATION RATE COEFFICIENT (CM+3 SEC-1)
C
C NOTES: THIS ROUTINE IS NOT YET PROPERLY ANNOTATED
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 16-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST VERSION
C
C VERSION: 1.2                          DATE: 16-05-07
C MODIFIED: Allan Whiteford
C             - Updated comments as part of subroutine documentation
C               procedure.
C
C-----------------------------------------------------------------------
C
      INTEGER             INTD,        N,           N11
      REAL*8              E,           E11,         EM,          PHI
      REAL*8              PY,          RDEXC,       TE,          WI
      REAL*8              WJ,          Z1
```

4.137 pyvr: Subroutine pyvr from library adas3xx

```
       SUBROUTINE PYVR(Y,Z1,PY)
C
       IMPLICIT REAL*8(A-H,O-Z)
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: PYVR ***********************
C
C  PURPOSE: CALCULATES VAN REGEMORTER'S P FACTOR FOR ELECTRON
C           COLLISIONS WITH ATOMS AND IONS.
C
C  INPUT
C      Y=ATE*(1/V1**2+1/V2**2)  WHERE
C                ATE=1.5789D5*Z1*Z1/TE
C                TE=ELECTRON TEMPERATURE (K)
C                V1=INITIAL EFFECTIVE PRINCIPAL QUANTUM NUMBER
C                V2=FINAL EFFECTIVE PRINCIPAL QUANTUM NUMBER
C      Z1=TARGET ION CHARGE+1
C      PY=P-FACTOR
C
C NOTES: THIS ROUTINE IS NOT YET PROPERLY ANNOTATED
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 16-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST VERSION
C
C VERSION: 1.2                          DATE: 16-05-07
C MODIFIED: Allan Whiteford
C             - Updated comments as part of subroutine documentation
C               procedure.
C-----------------------------------------------------------------------
C
       IF(Y-0.04)9,9,10
    9  PY=-0.2756644*(DLOG(Y)+0.41)
       GO TO 15
   10  IF(Z1-1.0)11,11,14
   11  IF(Y-1.0)12,12,13
   12  PY=(0.0196*Y+0.0882)/(Y+0.075)
       GO TO 15
   13  Y1=0.066/DSQRT(Y)
       PY=Y1*(1.0+8.0*Y1)
       GO TO 15
   14  Y1=1.0/Y
       PY=0.2+Y1*(0.04-0.00068*Y1)
   15  RETURN
       END
      REAL*8              PY,          Y,           Z1
```

4.138 qh: Subroutine qh from library adas3xx

```
       FUNCTION QH(EPRO,TTAR,ISEL,ZSEL,NSEL,IORD,EA,OA,N,IPASS,
     &             TITLF, DSLPATH)
C
       IMPLICIT REAL*8(A-H,O-Z)
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 FUNCTION: QH ***************************
C
C  PURPOSE : FUNCTION TO EVALUATE MAXWELL AVERAGED TOTAL
C      IONISATION, EXCITATION AND CHARGE EXCHANGE RATE
C      COEFFICIENTS. THE FUNCTION ALSO RETURNS THE RAW
C      CROSS-SECTION DATA FOR VERIFICATION AND GRAPHING
C      PURPOSES. THE INCIDENT PARTICLE IS A MONOENERGETIC
C      BEAM AND THE TARGET IS A MAXWELL DISTRIBUTION. THE
C      TARGET AND PROJECTILE ROLES MAY BE REVERSED. ARBITRARY
C      RELATIVE SPEEDS ARE ALLOWED. THE RATE COEFFICIENT
C      REQUIRED IS SELECTED FROM A LIST.  THERE IS SOME
C        SCALED DATA. ALL X-SECTS INVOLVE HYDROGEN AS ONE OF
C      THE SPECIES.
C
C  INPUT
C      EPRO  = INCIDENT PARTICLE ENERGY (EV/AMU)
C      TTAR  = MAXWELL TEMPERATURE OF TARGET PARTICLES (EV)
C              IF (TTAR.LE.0) THEN ONLY RAW SOURCE VALUES ARE RETURNED
C              IN ARRAYS (EA(I),OA(I), I=1,N)
C      ISEL  = SELECTOR FOR PARTICULAR RATE COEFFT. CHOSEN FROM TABLE
C              BELOW (SEE ALSO NOTES ON DATA)
C      ZSEL  = NUCLEAR CHARGE (REQUIRED ONLY FOR PARTICULAR ISEL)
C      NSEL  = PRINC. QUANTUM NO. (REQUIRED ONLY FOR PARTICULAR ISEL
C              NB. NSEL SHOULD BE ZERO ON ENTRY OTHERWISE)
C      IORD  = 1 FOR 1ST PARTICLE INCIDENT AND MONOENERGETIC
C            = 2 FOR 2ND PARTICLE INCIDENT AND MONOENERGETIC
C      IPASS = 0 IF DATA FILE TO BE READ IN AFRESH
C            = 1 IF DATA FILE IS NOT TO BE READ IN AGAIN
C  OUTPUT
C      QH    = RATE COEFFICIENT (CM3 SEC-1)
C      EA(I) = SET OF ENERGIES (EV/AMU) FOR SELECTED SOURCE DATA
C      OA(I) = CROSS-SECTIONS (CM**2) FOR SELECTED SOURCE DATA
C      N     = NUMBER OF SOURCE DATA VALUES
C      TITLF = INFORMATION STRING
C
C
C  *********** H.P.SUMMERS,JET             20 JUL 1990  ***************
C  ***********                       COR  31 JUL 1990  ***************
C  ***********                       ADD   3 JUL 1991  ***************
C
C NOTES: THIS ROUTINE IS NOT YET PROPERLY ANNOTATED
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 18-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - PUT UNDER SCCS CONTROL
C - MODIFIED NAME OF INPUT FILE BY ADDING INPUT
C                 PARAMETER DSLPATH CONTAINING THE PATH TO THE
C                 FILE AND INCREASING THE LENGTH OF THE VARIABLE DSNAME
C
C VERSION: 1.2                          DATE: 18-1-96
```

```
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - MODIFIED CONSTRUCTION OF DSLPATH
C
C VERSION: 1.3                          DATE: 22-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - REPLACED CALLS TO NAG ROUTINE E02BBF WITH CALL
C                 TO ADAS REPLACEMENT ROUTINE DXNBBF
C
C VERSION: 1.4                          DATE: 23-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - REPLACED CALL TO NAG ROUTINE E01BAF WITH CALL
C                 TO ADAS REPLACEMENT ROUTINE DXNBAF (NOTE THAT
C                 THIS IS NOT A DIRECT REPLACEMENT ROUTINE BUT
C                 HOPEFULLY IT WILL NOW GIVE THE SAME RESULTS)
C
C VERSION: 1.5                          DATE: 08-02-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - REMOVED SUPERFLUOUS VARIABLES
C
C VERSION: 1.6                          DATE: 08-02-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - CORRECTED NAME "TRING" TO "STRING"
C
C VERSION: 1.7   DATE: 26/11/96
C MODIFIED: HARVEY ANDERSON
C  MODIFIED FORMAT STATEMENT 1001 AND THE READ
C  STATEMENT BEGINING ON LINE 146 TO ACCOMODATE
C  FOR THE NEW FORMAT OF THE ADF02 FILE
C  /SIA#H/SIA#H_RFM#H.DAT. ALSO MODIFIED THE
C  DSNAME ON LINE 143. ALSO SLIGHTLY MODIFIED
C  LINE 152 AND 153 DUE TO THE NEW FORMAT OF
C  THE FILE. ALSO MODIFIED THE SELECT NUMBERS
C  ASSOCIATED WITH THE SCALING.
C
C VERSION: 1.8 DATE: 03-04-97
C MODIFIED: HARVEY ANDERSON
C  MODIFIED TO USE NEW PREFERRED ADF02 FILE sia#h_j97.dat
C
C VERSION: 1.9 DATE: 08-04-97
C MODIFIED: RICHARD MARTIN
C  CHANGED NAME OF ADF02 FILE FROM sia#h_j97.dat TO
C  sia#h_j97#h.dat
C
C VERSION: 1.10 DATE: 08-04-97
C MODIFIED: RICHARD MARTIN
C  CORRECTED MISTAKE MADE IN PREVIOUS MODIFICATION
C
C VERSION: 1.11 DATE: 09-08-99
C MODIFIED: HARVEY ANDERSON
C  CHANGED NAME OF THE ADF02 FILE FROM sia#h_j97#h.dat
C  TO sia#h_j99#h.dat.
C
C VERSION: 1.12 DATE: 20-10-2003
C MODIFIED: Martin O'Mullane
C  Save variables read from file between calls.
C
C VERSION: 1.13 DATE: 07-07-2004
C MODIFIED: ALLAN WHITEFORD
C  -CHANGED CALLS FROM DXNB{A,B}F TO XXNB{A,B}F
C
C-----------------------------------------------------------------------
```

```
CHARACTER*80        DSLPATH,     TITLF
INTEGER             IORD,        IPASS,       ISEL,        N
INTEGER             NSEL
REAL*8              EA(24),      EPRO,        OA(24),      TTAR
REAL*8              ZSEL
```

4.139 qhe: Subroutine qhe from library adas3xx

```
       FUNCTION QHE(EPRO,TTAR,ISEL,ZSEL,NSEL,IORD,EA,OA,N,IPASS,
     &              DSLPATH)
C
       IMPLICIT REAL*8(A-H,O-Z)
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 FUNCTION: QHE ***************************
C
C  PURPOSE : TO EVALUATE MAXWELL AVERAGED TOTAL IONISATION, EXCITATION
C      AND CHARGE EXCHANGE RATE COEFFICIENTS. FUNCTION ALSO
C      RETURNS THE RAW CROSS SECTION DATA FOR VERIFICATION AND
C      GRAPHING PURPOSES.
C            THE INCIDENT PARTICLE IS A MONOENERGETIC
C      BEAM AND THE TARGET IS A MAXWELL DISTRIBUTION. THE TARGET AND
C      PROJECTILE ROLES MAY BE REVERSED. ARBITRARY RELATIVE SPEEDS
C            ARE ALLOWED.
C
C  HISTORY : ROUTINE WAS ORIGINALLY WRITTEN BY H.P.SUMMERS AND MODIFIED
C      AND RESTRUCTURED BY HARVEY ANDERSON.
C
C  DATA   :  ROUTINE UTILISES THE ION/ATOM COLLISION DATABASE ASSEMBLED FOR
C      HELIUM IN INTERACTION WITH IMPURITIES UPTO THE SECOND
C      PERIOD. THE DATA BASE IS OF ADAS ADF02 TYPE FORMAT :
C
C /ADAS/ADF02/SIA#HE/SIA#HE_J91#HE.DAT
C
C  INPUT  :
C
C      (R*8) EPRO : INCIDENT PARTICLE ENERGY (EV/AMU).
C      (R*8) TTAR : MAXWELL TEMPERATURE OF TARGET
C PARTICLES (EV) IF (TTAR.LE.0) THEN
C ONLY RAW SOURCE VALUES ARE RETURNED
C               IN ARRAYS (EA(I),OA(I), I=1,N).
C      (R*8)      ZSEL : NUCLEAR CHARGE (REQUIRED ONLY FOR
C PARTICULAR ISEL).
C      (I*4) ISEL : SELECTOR FOR PARTICULAR RATE COEFFT.
C CHOSEN FROM TABLE BELOW (SEE ALSO
C NOTES ON DATA).
C      (I*4) NSEL : PRINC. QUANTUM NO. (REQUIRED ONLY FOR
C PARTICULAR ISEL NB. NSEL SHOULD BE
C ZERO ON ENTRY OTHERWISE).
C      (I*4)      IORD    : 1 FOR 1ST PARTICLE INCIDENT AND
C MONOENERGETIC = 2 FOR 2ND PARTICLE
C INCIDENT AND MONOENERGETIC.
C      (I*4) IPASS : 0 IF DATA FILE TO BE READ IN AFRESH
C             1 IF DATA FILE IS NOT TO BE READ IN
C AGAIN.
C
C
C  OUTPUT :
C            (R*8) QHE     : RATE COEFFICIENT (CM3 SEC-1).
C            (R*8)      EA :  SET OF ENERGIES (EV/AMU) FOR SELECTED
C SOURCE DATA.
C            (R*8)      OA      :       CROSS-SECTIONS (CM**2) FOR SELECTED
C SOURCE DATA.
C            (I*4)      N       :       NUMBER OF SOURCE DATA VALUES
C            (CHR) TITLF   : INFORMATION STRING.
C
```

```
C
C
C  CONTACT : HARVEY ANDERSON
C      UNIVERSITY OF STRATHCLYDE.
C      ANDERSON@CHAMBA.PHYS.STRATH.AC.UK
C
C  DATE    : 05/10/97
C
C
C VERSION: 1.2                          DATE: 15-10-99
C MODIFIED: RICHARD MARTIN
C   REMOVED 'ACTION' FROM OPEN STATEMENT.
C
C VERSION: 1.3 DATE: 07-07-2004
C MODIFIED: ALLAN WHITEFORD
C  -CHANGED CALLS FROM DXNB{A,B}F TO XXNB{A,B}F
C
C-----------------------------------------------------------------
      CHARACTER*80        DSLPATH
      INTEGER             IORD,        IPASS,       ISEL,        N
      INTEGER             NSEL
      REAL*8              EA(24),      EPRO,        OA(24),      TTAR
      REAL*8              ZSEL
```

4.140 qlpr: Subroutine qlpr from library adas3xx

```
       FUNCTION QLPR(Z1,N1,N2,E1,ZP,ATMSSP)
C
       IMPLICIT REAL*8(A-H,O-Z)
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 FUNCTION: QLPR ************************
C
C-----------------------------------------------------------------------
C  PURPOSE: CALCULATE LODGE-PERCIVAL-RICHARDS ION IMPACT EXCITATION
C           CROSS-SECTIONS IN ORIGINAL FORM  (J.PHYS.B. (1976)9,239).
C
C  EXCITATION CROSS-SECTION IS EVALUATED AND DE-EXCITATION CROSS-SECTION
C  OBTAINED BY DETAILED BALANCE
C
C  SCALING TO ARBITRARY PROJECTILE CHARGE FOLLOWS RECOMMENDATIONS
C  OF RIENHOLD,, OLSEN & FRITSCH (1990)PHYS.REV.A 41,4837
C
C
C  INPUT
C      Z1=TARGET ION CHARGE +1
C      N1=INITIAL PRINCIPAL QUANTUM NUMBER
C      N2=FINAL PRINCIPAL QUANTUM NUMBER
C      E1=ENERGY OF EQUIVALENT ELECTRON IN RYDBERGS
C         (CORRESPONDS TO ACTUAL PROJECTILE ENERGY/25KEV)
C      ZP=PROJECTILE CHARGE
C      ATMSSP= PROJECTILE MASS IN PROTON UNITS
C  OUTPUT
C      QLPR=CROSS-SECTION IN PI*A0**2 UNITS
C
C
C  ***********  H.P.SUMMERS, JET            16/ 7/90 ***************
C
C NOTES: THIS ROUTINE IS NOT YET PROPERLY ANNOTATED
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 16-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST VERSION
C
C VERSION: 1.2                          DATE: 16-05-07
C MODIFIED: Allan Whiteford
C             - Updated comments as part of subroutine documentation
C               procedure.
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
       ZZ1=Z1*Z1
       ZZP=ZP*ZP
       IF(N1.LT.N2)THEN
           E=E1
           T1=1.0D0
           EN1=N1
           EN2=N2
       ELSEIF (N1.EQ.N2)THEN
           QLPR=0.0D0
           RETURN
       ELSE
```

```
            E2=E1-ZZ1*(1.0D0/(EN1*EN1)-1.0D0/(EN2*EN2))/(1836.12*ATMSSP)
            E=E2
            T1=(EN2*EN2*E2)/(EN1*EN1*E1)
            EN1=N2
            EN2=N1
          ENDIF
          S=EN2-EN1
          EN12=EN1*EN2
          A=2.666667D0/S*(EN2/(S*EN1))**3*(0.184D0-0.04/S**0.66667D0)*
     &    (1.0D0-0.2D0*S/EN12)**(1.0D0+2.0D0*S)
          D=DEXP(-ZZ1*ZZP/(EN12*E*E))
          F=(1.0D0-0.3D0*S*D/EN12)**(1.0D0+2.0D0*S)
          Y=1.0D0/(1.0D0-D*DLOG(18.0D0*S)/(4.0D0*S))
          XL=DLOG((1.0D0+0.53D0*E*E*EN1*(EN2-2/EN2)/(ZZ1*ZZP))
     &                                  /(1.0D0+0.4D0*E/ZP))
          G=0.5D0*(E*EN1*EN1/(Z1*ZP*(EN2-1.0D0/EN2)))**3
          T=DSQRT(2.0D0-(EN1/EN2)**2)
          XP=2.0D0*Z1*ZP/(E*EN1*EN1*(T+1.0D0))
          XM=2.0D0*Z1*ZP/(E*EN1*EN1*(T-1.0D0))
          CP=(XP*XP/(2.0D0*Y+1.5D0*XP))*DLOG(1.0D0+0.66667D0*XP)
          CM=(XM*XM/(2.0D0*Y+1.5D0*XM))*DLOG(1.0D0+0.66667D0*XM)
          H=CM-CP
C         WRITE(6,1000)E,EN1,EN2,Z1,ZP,T1
C         WRITE(6,1001)A,D,XL,F,G,H
C         WRITE(7,1000)E,EN1,EN2,Z1,ZP,T1
C         WRITE(7,1001)A,D,XL,F,G,H
          QLPR=T1*EN1**4*(A*D*XL+F*G*H)*(ZZP/ZZ1)/E
          RETURN
 1000     FORMAT(1H ,'E=',1PD10.2,3X,0P,'EN1=',F4.1,3X,'EN2=',F4.1,3X,
     &           'Z1=',F4.1,3X,'ZP=',F4.1,3X,'T1=',1PD10.2)
 1001     FORMAT(1H ,'A=',1PD10.2,3X,'D=',1PD10.2,3X,'XL=',1PD10.2,3X,
     &           'F=',1PD10.2,3X,'GP=',1PD10.2,3X,'H=',1PD10.2)
          END
         INTEGER             N1,          N2
         REAL*8              ATMSSP,      E1,          Z1,          ZP
```

4.141 qpr78: Subroutine qpr78 from library adas3xx

```
       FUNCTION QPR78(Z1,N1,N2,E1,PHI)
C
       IMPLICIT REAL*8(A-H,O-Z)
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 FUNCTION: QPR78 ************************
C
C-----------------------------------------------------------------------
C  PURPOSE: CALCULATES ELECTRON COLLISION CROSS-SECTIONS FOR
C           TRANSITIONS BETWEEN PRINCIPAL QUANTUM SHELLS IN
C           HYDROGEN AND HYDROGENIC IONS.
C
C  PERCIVAL AND RICHARDS-MNRAS(1978)183,329.
C
C  VALID FOR INCIDENT ELECTRON ENERGIES IN RANGE (Z1/N1)**2<E1<137**2
C  AND FOR N1,N2>4.
C  ANOMALIES DEVELOP IN ORIGINAL SPECIFICATION OF PERCIVAL RICHARDS FOR
C  S=N2-N1 LARGE OR IF N1 IS <5.  HENCE A MODIFIED PRESCRIPTION IS USED
C  IN THESE CASES CONSISTENT WITH BANKS ET AL (1973) ASTR. LETT 14,161
C
C
C  INPUT
C      Z1=TARGET ION CHARGE +1
C      N1=INITIAL PRINCIPAL QUANTUM NUMBER
C      N2=FINAL PRINCIPAL QUANTUM NUMBER
C      E1=ENERGY OF INCIDENT ELECTRON IN RYDBERGS
C
C  OUTPUT
C      QPR78=CROSS-SECTION IN PI*A0**2 UNITS
C
C
C  ***********  H.P.SUMMERS, JET          12 NOV 1984        ***********
C  ****                               COR  28 FEB 1990               ****
C-----------------------------------------------------------------------
C NOTES: THIS ROUTINE IS NOT YET PROPERLY ANNOTATED
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 16-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST VERSION
C
C VERSION: 1.2                          DATE: 16-05-07
C MODIFIED: Allan Whiteford
C             - Updated comments as part of subroutine documentation
C               procedure.
C
C-----------------------------------------------------------------------
C
       ZZ1=Z1*Z1
       E=E1
       EN1=N1
       EN2=N2
       ABET=4.0D0*ZZ1*PHI/(EN1**4)
       T1=1.0D0
       IF(N1-N2)3,1,2
    1  QPR78=0.0D0
       RETURN
```

```
     2  E2=E1-ZZ1*(1.0D0/(EN1*EN1)-1.0D0/(EN2*EN2))
        T1=(EN2*EN2*E2)/(EN1*EN1*E1)
        EN1=N2
        EN2=N1
        E=E2
     3  S=EN2-EN1
        EN12=EN1*EN2
        IF(N1.LT.5.OR.S.GT.10.0D0)GO TO 4
        A=2.666667D0/S*(EN2/(S*EN1))**3*(0.184D0-0.04/S**0.66667D0)*(1.0
       &D0-0.2D0*S/EN12)**(1.0D0+2.0D0*S)
        D=DEXP(-ZZ1/(EN12*E*E))
        F=(1.0D0-0.3D0*S*D/EN12)**(1.0D0+2.0D0*S)
        Y=1.0D0/(1.0D0-D*DLOG(18.0D0*S)/(4.0D0*S))
        GO TO 5
     4  A=ABET
        D=DEXP(-Z1/(EN1*E))
C       D=DEXP(-ZZ1/(EN12*E*E))
        F=1.0D0-D
        Y=1.0D0
     5  XL=DLOG((1.0D0+0.53D0*E*E*EN12/ZZ1)/(1.0D0+0.4D0*E))
        G=0.5D0*(E*EN1*EN1/(Z1*EN2))**3
        T=DSQRT(2.0D0-(EN1/EN2)**2)
        XP=2.0D0*Z1/(E*EN1*EN1*(T+1.0D0))
        XM=2.0D0*Z1/(E*EN1*EN1*(T-1.0D0))
        CP=(XP*XP/(2.0D0*Y+1.5D0*XP))*DLOG(1.0D0+0.66667D0*XP)
        CM=(XM*XM/(2.0D0*Y+1.5D0*XM))*DLOG(1.0D0+0.66667D0*XM)
        H=CM-CP
        QPR78=T1*EN1**4*(A*D*XL+F*G*H)/(E*ZZ1)
        RETURN
        END
       INTEGER             N1,          N2
       REAL*8              E1,          PHI,         Z1
```

4.142 qvain: Subroutine qvain from library adas3xx

```
       FUNCTION QVAIN(Z1,N1,N2,VION,PHI,ZP)
C
       IMPLICIT REAL*8(A-H,O-Z)
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 FUNCTION: QVAIN ************************
C
C  PURPOSE: CALCULATES ION COLLISION CROSS-SECTIONS FOR TRANSITIONS
C           BETWEEN PRINCIPAL QUANTUM SHELLS IN HYDROGEN AND
C           HYDROGENIC IONS.
C
C  INPUT
C      Z1   = TARGET ION CHARGE +1
C      N1   = INITIAL PRINCIPAL QUANTUM NUMBER
C      N2   = FINAL PRINCIPAL QUANTUM NUMBER
C      VION = VELOCITY OF INCIDENT ION (CM/SEC)
C      ZP   = PROJECTILE CHARGE
C  OUTPUT
C      QVAIN=CROSS-SECTION IN PI*A0**2 UNITS
C
C NOTES: THIS ROUTINE IS NOT YET PROPERLY ANNOTATED
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 16-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST VERSION
C
C VERSION: 1.2                          DATE: 16-05-07
C MODIFIED: Allan Whiteford
C             - Updated comments as part of subroutine documentation
C               procedure.
C-----------------------------------------------------------------------
C
       ZZ1=Z1*Z1
       ZZP=ZP*ZP
       EN1=N1
       EN2=N2
       EIJRH=DABS(1/(EN1*EN1) - 1/(EN2*EN2))
       ALFA=1.0/137.036
       CLIG=2.998D+10
       CFAC=(CLIG/VION)**2
       BETA=0.5*ALFA*ALFA*CFAC*DSQRT(PHI)*EIJRH*ZP
       XFAC1=2*ALFA*ALFA*CFAC*PHI*ZP*ZP
       QVAIN=XFAC1*DEXP(-2*DSQRT(BETA))*RIFAC(BETA)
       RETURN
       END
C
       FUNCTION RIFAC(BETAIN)
       IMPLICIT REAL*8(A-H,O-Z)
      INTEGER             N1,          N2
      REAL*8              PHI,         VION,        Z1,          ZP
      REAL*8              BETAIN
```

4.143 rd2bs: Subroutine rd2bs from library adas3xx

```
       FUNCTION RD2BS(N,L,N2,L2)
       IMPLICIT REAL*8(A-H,O-Z)
C-----------------------------------------------------------------------
C
C  PURPOSE: GENERATION OF HYDROGENIC BOUND-BOUND RADIAL INTEGRALS USING
C           RECURRENCE RELATIONS.
C
C-----------------------------------------------------------------------
C VERSION  : 1.1
C DATE     : 18-03-1999
C MODIFIED : ???
C
C VERSION  : 1.2
C DATE     : 05-10-2000
C MODIFIED : ???
C             - Removed junk from columns > 72
C
C VERSION  : 1.3
C DATE     : 16-05-2007
C MODIFIED : Allan Whiteford
C             - Updated comments as part of subroutine documentation
C               procedure.
C
C-----------------------------------------------------------------------
       SC=64.0
       SCL=0.015625
       EN=N
       EN2=EN*EN
       EK=N2
       EK=1.0/EK
       EK2=-EK*EK
       V=1.0+EN2*EK2
       W=1.0+EN*EK
       W=W*W
       U=8.0*EN2/(W*W)
       P=1.0
       JS=0
       SC2=SC*SC
       SCL2=SCL*SCL
       DO 5 I=1,N
       EI=I
       P=P*U*(1.0+EI*EI*EK2)/(EI*(2.0*EI-1.0))
       AP=DABS(P)
       IF(SCL2.LE.AP)GO TO 5
       JS=JS-1
       P=SC2*P
    5  CONTINUE
       P=2.0*EN*EK*P
       P=4.0*EK*DSQRT(P)/(V*V)
       NL=N+1
       DO 10 I=NL,N2
       P=P*V/W
       AP=DABS(P)
       IF(SCL.LE.AP)GO TO 10
       JS=JS-1
       P=SC*P
   10  CONTINUE
       IF(L2.EQ.L+1)GO TO 11
       IF(L2.EQ.L-1)GO TO 20
```

```
    RD2BS=0.0
    GO TO 7
11  T2=P
    U=(2.0*EN-1.0)*V
    U=DSQRT(U)
    T3=0.5*U*T2
    NU=N-2
    IF(L-NU)14,13,12
12  T3=T2
13  GO TO 40
14  DO 16 I=L2,NU
    LI=NU-I+L
    EL1=LI+1
    EL2=LI+2
    ES=EL2*EL2
    T1=T2
    T2=T3
    T3=(4.0*(EN2-ES)+EL2*(2.0*EL2-1.0)*V)*T2-2.0*EN*U*T1
    U=(EN2-EL1*EL1)*(1.0+ES*EK2)
    U=DSQRT(U)
    T3=T3/(2.0*EN*U)
    AT3=DABS(T3)
    IF(AT3.LE.SC)GO TO 16
    JS=JS+1
    T3=SCL*T3
    T2=SCL*T2
16  CONTINUE
    GO TO 40
20  T2=P
    EN1=N-1
    U=V/(1.0+EN1*EN1*EK2)
    T2=DSQRT(U)*T2/(2.0*EN)
    U=(2.0*EN-1.0)*(1.0+(EN-2.0)*(EN-2.0)*EK2)
    U=DSQRT(U)
    T3=(4.0+EN1*V)*(2.0*EN-1.0)*T2/(2.0*EN*U)
    NU=N-3
    IF(L-NU-1)24,23,22
22  T3=T2
23  GO TO 40
24  DO 26 I=L,NU
    LI=NU-I+L
    EL=LI
    EL1=LI+1
    ES=EL1*EL1
    T1=T2
    T2=T3
    T3=(4.0*(EN2-ES)+EL1*(2.0*EL1+1.0)*V)*T2-2.0*EN*U*T1
    U=(EN2-ES)*(1.0+EL*EL*EK2)
    U=DSQRT(U)
    T3=T3/(2.0*EN*U)
    AT3=DABS(T3)
    IF(AT3.LE.SC)GO TO 26
    JS=JS+1
    T3=SCL*T3
    T2=SCL*T2
26  CONTINUE
40  RD2BS=EN2*EN2*T3*T3*4096.0**JS
 7  RETURN
   END
   INTEGER             L,           L2,          N,           N2
```

4.144 rndegv: Subroutine rndegv from library adas3xx

```
      SUBROUTINE RNDEGV(Z0,Z,ZEFF1,ZEFF2,NI,LI,LTI,EI,WI,NJ,LJ,LTJ,EJ,
     &WJ,IS,ISP,LP,N0,TEV,DENS,TPV,ZP,EMP,GAE,GAP)
      IMPLICIT REAL*8(A-H,O-Z)
C-----------------------------------------------------------------------
C  PURPOSE: CALCULATES TERM AVERAGED ELECTRON AND +VE ION COLLISIONAL
C           RATE PARAMETERS BETWEEN NEARLY DEGENERATE LEVELS.
C
C  EXPLICIT ENERGY LEVEL CALCULATIONS ARE PERFORMED FOR H, HE AND LI-
C  LIKE SYSTEMS AND A SIMPLE SUM OF J RESOLVED COMPONENT RATE
C  PARAMETERS IS FORMED. FOR OTHER SYSTEMS THE INPUT ENERGIES ARE USED.
C  THE DRIVING PROGRAM REQUIRES THE STATEMENT
C      CALL GAMAF(200)
C  PRIOR TO CALLS TO RNDEGV
C  ******************** H.P. SUMMERS, JET   18 FEB 1985 ***************
C  ***** CORRECTIONS 22/5/85
C  INPUT
C      Z0=TARGET NUCLEAR CHARGE
C      Z=TARGET ION CHARGE
C      ZEFF1=TARGET EFFECTIVE CHARGE (FOR CORRELATION PART OF ENERGY)
C      ZEFF2=TARGET EFFECTIVE CHARGE (FOR RELATIVISITIC PART OF ENERGY)
C      NI=VALENCE ELECTRON PRINCIPAL QUANTUM NUMBER IN STATE I
C      LI=VALENCE ELECTRON ORBITAL QUANTUM NUMBER IN STATE I
C      LTI=TOTAL ORBITAL ANGULAR MOMENTUM IN STATE I
C      EI=BINDING ENERGY OF STATE I (RYD)
C      WI=STATISTICAL WEIGHT OF STATE I (FULL WEIGHTING INCLUDING SPIN)
C      NJ=VALENCE ELECTRON PRINCIPAL QUANTUM NUMBER IN STATE J
C      LJ=VALENCE ELECTRON ORBITAL QUANTUM NUMBER IN STATE J
C      LTJ=TOTAL ORBITAL ANGULAR MOMENTUM IN STATE J
C      EJ=BINDING ENERGY OF STATE J (RYD)
C      WJ=STATISTICAL WEIGHT OF STATE J
C      IS=(2*S+1) WHERE S IS TOTAL SPIN
C      ISP=(2*SP+1) WHERE SP IS PARENT SPIN
C      LP=PARENT ORBITAL ANGULAR MOMENTUM
C      N0=LOWEST PRINCIPAL QUANTUM NUMBER FOR ION (FOR LIFETIME)
C      TEV=TEMPERATURE(EV) (ELECTRON DISTRIBUTION)
C      DENS=ELECTRON DENSITY (CM-3)
C      TPV=TEMPERATURE(EV) (COLLIDING +VE ION DISTRIBUTION)
C      ZP=CHARGE OF COLLIDING +VE ION
C      EMP=REDUCED MASS FOR COLLIDING +VE ION (ELECTRON MASSES)
C  OUTPUT
C      GAE=GAMMA RATE PARAMETER FOR ELECTRON COLLISIONS
C      GAP=GAMMA RATE PARAMETER FOR +VE ION COLLISIONS
C-----------------------------------------------------------------------
C VERSION  : 1.1
C DATE     : 18-03-1999
C MODIFIED : ???
C
C VERSION  : 1.2
C DATE     : 05-10-2000
C MODIFIED : ???
C             - Removed junk from columns > 72
C
C VERSION  : 1.3
C DATE     : 16-05-2007
C MODIFIED : Allan Whiteford
C             - Updated comments as part of subroutine documentation
C               procedure.
C
C-----------------------------------------------------------------------
```

```
INTEGER            IS,          ISP,         LI,          LJ
INTEGER            LP,          LTI,         LTJ,         N0
INTEGER            NI,          NJ
REAL*8             DENS,        EI,          EJ,          EMP
REAL*8             GAE,         GAP,         TEV,         TPV
REAL*8             WI,          WJ,          Z,           Z0
REAL*8             ZEFF1,       ZEFF2,       ZP
```

4.145 rpengv: Subroutine rpengv from library adas3xx

```
       SUBROUTINE RPENGV(IZ,WI,EI,WJ,EJ,N,LI,LJ,PHI,TV,TEV,DENS,TAU,EM,
     &IZC,QI,QJ,GA)
       IMPLICIT REAL*8(A-H,O-Z)
C-----------------------------------------------------------------------
C  PURPOSE: CALCULATES PENGELLY & SEATON (1964) COLLISION RATES BETWEEN
C           NEARLY DEGENERATE LEVELS.
C
C  A VARIATION OF IMPACT PARAMETER THEORY FOR DIPOLE TRANSITIONS IS USED.
C  ************** H.P.SUMMERS,JET   2 DEC 1984  **********************
C  *** CORRECTIONS 13/5/85
C  INPUT
C      IZ=TARGET ION CHARGE
C      WI=STATISTICAL WEIGHT OF STATE I (FULL WEIGHTING INCLUDING SPIN)
C      EI=BINDING ENERGY OF STATE I (RYD)
C      WI=STATISTICAL WEIGHT OF STATE J
C      EJ=BINDING ENERGY OF STATE J (RYD)
C      PHI=FIJ/EIJ (=SIJ/WI) WITH FIJ=ABSORPTION OSCILLATOR STRENGTH
C                                 EIJ=TRANSITION ENERGY (RYDBERGS)
C                                 SIJ=LINE STRENGTH (AT. UNITS)
C      TV=TEMPERATURE(EV)   (COLLIDING PARTICLE DISTRIBUTION)
C      TEV=TEMPERATURE(EV)  (ELECTRON DISTRIBUTION)
C      DENS=ELECTRON DENSITY (CM-3)
C      TAU=MEAN RADIATIVE LIFETIME OF INITIAL AND FINAL LEVELS (SEC)
C      EM=REDUCED MASS FOR COLLIDING PARTICLE (ELECTRON MASSES)
C      IZC=CHARGE OF COLLIDING PARTICLE
C  OUTPUT
C      QI=EXCITATION RATE COEFFICIENT (CM**3 SEC-1)
C      QJ=DEEXCITATION RATE COEFFICIENT
C      GA=GAMMA RATE PARAMETER
C-----------------------------------------------------------------------
C VERSION  : 1.1
C DATE     : 18-03-1999
C MODIFIED : ???
C
C VERSION  : 1.2
C DATE     : 05-10-2000
C MODIFIED : ???
C             - Removed junk from columns > 72
C
C VERSION  : 1.3
C DATE     : 16-05-2007
C MODIFIED : Allan Whiteford
C             - Updated comments as part of subroutine documentation
C               procedure.
C
C-----------------------------------------------------------------------
       T=1.16054D4*TV
       TE=1.16054D4*TEV
       ATP=1.5789D5/T
       Z1=IZ+1
       ZC=IZC
       XN=N
       XLI=LI
       XLJ=LJ
       XL=0.5D0*(XLI+XLJ)
       DNL=6.0D0*(ZC*XN/Z1)**2*(XN*XN-XL*XL-XL-1.0D0)
       EIJ=DABS(EI-EJ)
       TAU1=1.0D10
       IF(EIJ.GT.0.0D0)TAU1=7.53D-17/EIJ
```

```
        IF(TAU1-TAU)3,3,2
     2  TAU1=TAU
        IND1=0
        GO TO 4
     3  IND1=1
C  IND1=0 INDICATES FINITE RADIATIVE LIFETIME CUT-OFF
C       =1 INDICATES BETHE CUT-OFF
     4  F1=1.68+DLOG10(TE/DENS)
        F=10.95+DLOG10(T*TAU1*TAU1/EM)
C**     WRITE(6,100)TAU,TAU1,IND1,F,F1
  100   FORMAT(1H0,'CHECK OUTPUT FROM RPENGV'/
     &1H ,'TAU =',1PD10.2,3X,'TAU1 =',1PD10.2,3X,'IND1 =',I3,3X,
     &'F =',1PD10.2,3X,'F1 =',1PD10.2)
        IF(F-F1)8,8,9
     8  IND2=0
        GO TO 10
     9  F=F1
        IND2=1
C  IND2=0 INDICATES LIFETIME OR BETHE CUTOFF USED IN RATE
C       =1 INDICATES DEBYE CUT-OFF USED IN RATE
    10  B=11.54+DLOG10(T/(DNL*EM))+F
C**     WRITE(6,101)EIJ,DNL,T,EM,B,F
  101   FORMAT(1H ,'EIJ =',1PD10.2,3X,'DNL =',1PD10.2,3X,'T =',1PD10.2,
     &3X,'EM =',1PD10.2,3X,'B =',1PD10.2,3X,'F =',1PD10.2)
        IF(B-1.0D0)14,14,15
    14  IF(B.GT.0.0D0.AND.IND2.EQ.1)GO TO 15
        IF(B.GT.0.0D0.AND.IND1.EQ.0)GO TO 15
        QI=0.0D0
        GO TO 16
    15  QI=7.94D-5*DSQRT(EM/T)*ZC*ZC*PHI*B
    16  QJ=WI*QI/WJ
        GA=4.604D7*WJ*QJ/DSQRT(ATP)
        RETURN
       END
       INTEGER             IZ,          IZC,         LI,          LJ
       INTEGER             N
       REAL*8              DENS,        EI,          EJ,          EM
       REAL*8              GA,          PHI,         QI,          QJ
       REAL*8              TAU,         TEV,         TV,          WI
       REAL*8              WJ
```

4.146 rqinew: Subroutine rqinew from library adas3xx

```
       FUNCTION RQINEW(Z,N,ZIMP,AMSIMP,TP,VDISP)
C
       IMPLICIT REAL*8(A-H,O-Z)
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 FUNCTION: RQINEW ***********************
C
C-----------------------------------------------------------------------
C  PURPOSE: EVALUATES ION IMPACT IONISATION RATE COEFFICIENTS FOLLOWING
C           THE EXPRESSIONS OF PERCIVAL AND RICHARDS.
C
C  (ALTERNATIVE TO RQIONPR WITH BETTER MAXWELL AVERAGING BUT SLOWER)
C
C  A CONSTANT SPEED SHIFT MAY BE GIVEN TO THE COLLISION OVER AND ABOVE
C  THE THERMAL SPEEDS
C
C
C  INPUT
C      Z      = TARGET ION CHARGE+1
C      N      = PRINCIPAL QUANTUM  NUMBER OF INITIAL  TARGET LEVEL
C      ZIMP     = PROJECTILE CHARGE
C      AMSIMP = PROJECTILE MASS (PROTON UNITS)
C      TP     = ION TEMPERATURE (K)    (EITHER TARGET OR PROJECTILE)
C      VDISP  = CONSTANT MEAN SPEED SHIFT FOR THE COLLISION
C                     (DESCRIBES BEAM PLASMA SITUATIONS)
C
C  OUTPUT
C      RQINEW  = RATE COEFFICIENT  (CM**3 SEC-1)
C
C
C  ********* H.P. SUMMERS, JET               3  JULY  1991   ***********
C
C NOTES: THIS ROUTINE IS NOT YET PROPERLY ANNOTATED
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 16-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST VERSION
C
C VERSION: 1.2                          DATE: 08-02-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - REMOVED SUPERFLUOUS VARIABLES
C
C VERSION: 1.3                          DATE: 16-05-07
C MODIFIED: Allan Whiteford
C             - Updated comments as part of subroutine documentation
C               procedure.
C
C-----------------------------------------------------------------------
C
      INTEGER             N
      REAL*8              AMSIMP,      TP,          VDISP,       Z
      REAL*8              ZIMP
```

4.147 rqlnew: Subroutine rqlnew from library adas3xx

```
       FUNCTION RQLNEW(Z,N11,N,PHI,ZP, AMSIMP,TP,VDISP)
C
       IMPLICIT REAL*8(A-H,O-Z)
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 FUNCTION: RQLNEW ***********************
C
C-----------------------------------------------------------------------
C  PURPOSE: EVALUATES ION IMPACT RATE COEFFICIENTS OF LODGE,
C           PERCIVAL & RICHARDS
C
C  (ALTERNATIVE TO RQLPR  WITH BETTER MAXWELL AVERAGING BUT SLOWER)
C
C  ROUTINE MUST RETURN EXCITATION RATE COEFFICIENT IF N11.LT.N   AND
C  DEXCITATION RATE COEFFICIENT IF N11.GT.N
C
C  NOTE THAT THE RELATION BETWEEN INVERSE PROCESSES IS DETERMINED BY THE
C  TEMPERATURE TP AND THE SPEED VDISP
C  THE TREATMENT GIVEN IS APPROXIMATE EXCEPT IN THE LIMITS VDISP=0 OR
C  VDISP >> DSQRT(2*TP/AMSIMP)
C
C  INPUT
C      Z      = TARGET ION CHARGE+1
C      N11    = PRINCIPAL QUANTUM  NUMBER OF INITIAL LEVEL
C      N      = PRINCIPAL QUANTUM  NUMBER OF FINAL   LEVEL
C      PHI    = (IH/EIJ)F(N ---> N'')
C      ZP     = PROJECTILE CHARGE
C      AMSIMP = PROJECTILE MASS (PROTON UNITS)
C      TP     = PROJECTILE ION TEMPERATURE (K)
C      VDISP  = CONSTANT MEAN SPEED SHIFT FOR THE COLLISION (CM/SEC)
C                     (DESCRIBES BEAM PLASMA SITUATIONS)
C
C  OUTPUT
C      RQLNEW = RATE COEFFICIENT (CM**3 SEC-1)
C
C
C  ********* H.P. SUMMERS, JET                2 JUL 1991   ************
C
C NOTES: THIS ROUTINE IS NOT YET PROPERLY ANNOTATED
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 16-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST VERSION
C
C VERSION: 1.2                          DATE: 08-02-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - REMOVED SUPERFLUOUS VARIABLES
C
C VERSION: 1.3                          DATE: 16-05-07
C MODIFIED: Allan Whiteford
C             - Updated comments as part of subroutine documentation
C               procedure.
C
C-----------------------------------------------------------------------
C
      INTEGER             N,           N11
```

```
      REAL*8            AMSIMP,      PHI,         TP,          VDISP
      REAL*8            Z,           ZP
```

4.148 rqvnew: Subroutine rqvnew from library adas3xx

```
       FUNCTION RQVNEW(Z,N11,N,PHI,ZP, AMSIMP,TP,VDISP)
C
       IMPLICIT REAL*8(A-H,O-Z)
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 FUNCTION: RQVNEW ***********************
C
C-----------------------------------------------------------------------
C  PURPOSE: EVALUATES ION IMPACT RATE COEFFICIENTS OF
C           VAINSHTEIN ET AL 1981
C
C  (ALTERNATIVE TO RQVAIN WITH BETTER MAXWELL AVERAGING BUT SLOWER)
C
C  ROUTINE MUST RETURN EXCITATION RATE COEFFICIENT IF N11.LT.N   AND
C  DEXCITATION RATE COEFFICIENT IF N11.GT.N
C
C  NOTE THAT THE RELATION BETWEEN INVERSE PROCESSES IS DETERMINED BY THE
C  TEMPERATURE TP AND THE SPEED VDISP
C  THE TREATMENT GIVEN IS APPROXIMATE EXCEPT IN THE LIMITS VDISP=0 OR
C  VDISP >> DSQRT(2*TP/AMSIMP)
C
C  INPUT
C      Z      = TARGET ION CHARGE+1
C      N11    = PRINCIPAL QUANTUM  NUMBER OF INITIAL LEVEL
C      N      = PRINCIPAL QUANTUM  NUMBER OF FINAL   LEVEL
C      PHI    = (IH/EIJ)F(N ---> N'')
C      ZP     = PROJECTILE CHARGE
C      AMSIMP = PROJECTILE MASS (PROTON UNITS)
C      TP     = PROJECTILE ION TEMPERATURE (K)
C      VDISP  = CONSTANT MEAN SPEED SHIFT FOR THE COLLISION (CM/SEC)
C                     (DESCRIBES BEAM PLASMA SITUATIONS)
C
C  OUTPUT
C      RQVNEW = RATE COEFFICIENT (CM**3 SEC-1)
C
C
C  ********* H.P. SUMMERS, JET                2 JUL 1991   ************
C
C NOTES: THIS ROUTINE IS NOT YET PROPERLY ANNOTATED
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 16-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST VERSION
C
C VERSION: 1.2                          DATE: 08-02-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - REMOVED SUPERFLUOUS VARIABLES
C
C VERSION: 1.3                          DATE: 16-05-07
C MODIFIED: Allan Whiteford
C             - Updated comments as part of subroutine documentation
C               procedure.
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
C
```

```
      INTEGER           N,          N11
      REAL*8            AMSIMP,     PHI,        TP,         VDISP
      REAL*8            Z,          ZP
```

4.149 sfi2: Subroutine sfi2 from library adas3xx

```
      SUBROUTINE SFI2

      IMPLICIT REAL*8 (A-H,O-Z)
C-----------------------------------------------------------------------
C     PURPOSE: This subroutine find the value ln(SIGMA(XNEW))
C-----------------------------------------------------------------------
C
C ADAS305 version. Developed from JETSHP.STARK.FORT (H P Summers).
C
C VERSION  : 1.1
C DATE     : 24-02-2005
C MODIFIED : Martin O'Mullane
C             - First version.
C
C VERSION  : 1.2
C DATE     : 16-05-07
C MODIFIED : Allan Whiteford
C             - Updated comments as part of subroutine documentation
C               procedure.
C-----------------------------------------------------------------------

      COMMON/APRX/F(10),B(10),H(10),ABETA(10)
      COMMON/B/FSPL(1),XNEW(1)
```

4.150 sigia: Subroutine sigia from library adas3xx

```
C
      SUBROUTINE SIGIA ( LSETX    , LPASS    ,
     &                   ALPH     , ETH      , ILTYP    , IOPT     ,
     &                   NENIN    , ENIN     , SGIN     ,
     &                   LTHETA   , VREL     , XSEC
     &                 )
C-----------------------------------------------------------------------
C
C  ********************** FORTRAN77 SUBROUTINE: SIGIA ******************
C
C  VERSION: 1.0 (ADAS91)
C
C  PURPOSE:  INTERPOLATES CROSS-SECTION DATA FROM AN INPUT VECTOR OF
C            VALUES USING CUBIC SPLINES.
C
C  EXTRAPOLATES FOR RELATIVE SPEEDS OUT OF DATA RANGE ACCORDING TO
C  VARIOUS TYPES (ILTYP).  LOGARITHMIC INTERPOLATION MAY BE USED
C  (LPASS). SPEED ECONOMY IS POSSIBLE FOR REPEATS WITH THE SAME
C  SPLINE KNOTS (LSETX).
C
C
C  CALLING PROGRAM:  CXTHER
C
C  NOTES:
C        (1) FOR  ILTYP.EQ.0, EXTRAPOLATION IS AS FOLLOWS:
C
C
C  SUBROUTINE:
C
C  INPUT : (L*4)   LSETX    = .TRUE. => SPLINE NOT SET FOR THESE KNOTS
C                             .FLSE. => SPLINE NOT FOR THESE KNOTS
C  INPUT : (L*4)   LPASS    = .TRUE. => DO NOT CONVERT INTO LOG10 FOR
C                                       ENERGIES AND X-SECTS. FOR SPLINE
C                             .FLSE. => CONVERT INTO LOG10 FOR
C                                       ENERGIES AND X-SECTS. FOR SPLINE
C  INPUT : (R*8)   ALPH     = HIGH ENERGY EXTRAPOLATION PARAMETER
C  INPUT : (R*8)   ETH      = THRESHOLD ENERGY (RYD.)
C  INPUT : (I*4)   ILTYP    = TYPE FOR LOW AND HIGH ENERGY CROSS-
C                             -SECTION EXTRAPOLATION.
C  INPUT : (I*4)   IOPT     = SPLINE END POINT CURVATURE/GRADIENT OPTION
C                             1 => DDY1 = 0, DDYN = 0
C                             4 => DY1 = 0 , DDYN = 0
C
C  INPUT : (I*4)   NENIN    = NUMBER OF ENERGIES IN INPUT DATA SET
C  INPUT : (R*8)   ENIN()   = ENERGIES (EV/AMU) IN INPUT DATA SET
C  INPUT : (R*8)   SGIN()   = INPUT X-SECTIONS (CM2) FROM INPUT DATA SET
C                             1ST.DIM: ENERGY INDEX
C  INPUT : (I*4)   LTHETA   = NUMBER OF VALUES IN VREL VECTOR
C  INPUT : (R*8)   VREL()   = RELATIVE SPEEDS FOR OUTPUT (CM S-1)
C
C  OUTPUT: (R*8)   XSEC()   = OUTPUT CROSS-SECTION (CM2)
C
C          (I*4)   MAXENS   = PARAMETER = MAX. LENGTH OF TABULAR XSECT.
C                                         VECTOR
C          (I*4)   LDTHET   = PARAMETER = MAX. LENGTH OF INTERNAL
C                                         VECTORS
C          (R*8)   CMSAMU   = PARAMETER = CONVERSION FACTOR FOR ENERGY
C                                         (AMU) TO VELOCITY (CM S-1)
C
```

```
C          (I*4)   I        = GENERAL INDEX
C          (I*4)   N        = GENERAL INDEX
C          (R*8)   ALPH0    = LOW VELOCITY EXTRAPOLATION PARAMETER
C          (R*8)   EXPON    = EXPONENT OF EXPONENTIAL
C          (R*8)   VSLOPE   = HIGH VELOCITY EXTRAPOLATION PARAMETER
C          (R*8)   XIN()    = INTERNAL SPLINE INDEPENDENT VARIABLE
C          (R*8)   YIN()    = INTERNAL SPLINE DEPENDENT VARIABLE
C          (R*8)   VIN()    = INTERNAL VECTOR
C          (R*8)   DY()     = DERIVATIVES AT SPLINE KNOTS
C          (R*8)   XOUT()   = INTERNAL OUTPUT INDEPENDENT VARIABLE
C          (R*8)   YOUT()   = INTERNAL OUTPUT DEPENDENT VARIABLE
C          (L*4)   LINTRP() = .TRUE.  => POINT INTERPOLATED
C                           = .FALSE. => POINT EXTRAPOLATED
C
C
C  ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          XXSPLE     ADAS      INTERPOLATES USING CUBIC SPLINES
C          R8FUN1     ADAS      EXTERNAL FUNCTION FOR XXSPLE
C
C
C  AUTHOR:  H. P. SUMMERS, UNIVERSITY OF STRATHCLYDE
C           JA8.08
C           TEL. 0141-553-4196
C
C  DATE:    18/11/96
C
C VERSION: 1.1 DATE: 18-11-96
C MODIFIED: HUGH P. SUMMERS, UNIVERSITY OF STRATHCLYDE
C     - FIRST EDITION
C
C VERSION: 1.1 DATE: 17-05-07
C MODIFIED: Allan Whiteford
C     - Modified comments as part of subroutine documentation
C             procedure.
C
C
C-----------------------------------------------------------------------
      INTEGER             ILTYP,       IOPT,        LTHETA,      NENIN
      LOGICAL             LPASS,       LSETX
      REAL*8              ALPH,        ENIN(NENIN), ETH
      REAL*8              SGIN(NENIN), VREL(LTHETA)
      REAL*8              XSEC(LTHETA)
```

4.151 sigma: Subroutine sigma from library adas3xx

```
      SUBROUTINE SIGMA

      IMPLICIT REAL*8 (A-H,O-Z)
C-----------------------------------------------------------------------
C
C     PURPOSE: Calculates <SIGMA*V> or SIGMA*V rates for collisional
C              excitation by proton or electron impact
C
C-----------------------------------------------------------------------
C
C ADAS305 version. Developed from JETSHP.STARK.FORT (H P Summers).
C
C VERSION  : 1.1
C DATE     : 24-02-2005
C MODIFIED : Martin O'Mullane
C             - First version.
C
C VERSION  : 1.2
C DATE     : 24-10-2005
C MODIFIED : Martin O'Mullane
C             - Change MK(66) to BK(66) to maintain maning and
C               size of ADAT common block.
C
C VERSION  : 1.3
C DATE     : 16-05-2007
C MODIFIED : Allan Whiteford
C             - Updated comments as part of subroutine documentation
C               procedure.
C
C-----------------------------------------------------------------------
```

4.152 sigmel: Subroutine sigmel from library adas3xx

```
      SUBROUTINE SIGMEL

      IMPLICIT REAL*8 (A-H,O-Z)
C-----------------------------------------------------------------------
C
C  PURPOSE: Calculates rate coefficients for excitation by
C           electron impact
C
C-----------------------------------------------------------------------
C
C ADAS305 version. Developed from JETSHP.STARK.FORT (H P Summers).
C
C VERSION  : 1.1
C DATE     : 24-02-2005
C MODIFIED : Martin O'Mullane
C             - First version.
C
C VERSION  : 1.2
C DATE     : 16-05-2007
C MODIFIED : Allan Whiteford
C             - Updated comments as part of subroutine documentation
C               procedure.
C
C-----------------------------------------------------------------------
```

4.153 sigmin: Subroutine sigmin from library adas3xx

```
      SUBROUTINE SIGMIN

      IMPLICIT REAL*8 (A-H,O-Z)
C-----------------------------------------------------------------------
C  PURPOSE: Calculates rate coefficients for excitation by ion impact
C-----------------------------------------------------------------------
C
C ADAS305 version. Developed from JETSHP.STARK.FORT (H P Summers).
C
C VERSION  : 1.1
C DATE     : 24-02-2005
C MODIFIED : Martin O'Mullane
C             - First version.
C
C VERSION  : 1.2
C DATE     : 16-05-2007
C MODIFIED : Allan Whiteford
C             - Updated comments as part of subroutine documentation
C               procedure.
C
C-----------------------------------------------------------------------
```

4.154 spij3: Subroutine spij3 from library adas3xx

```
       SUBROUTINE SPIJ3(N,H,W)
C
       IMPLICIT REAL*8(A-H,O-Z)
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: SPIJ3 **********************
C
C PURPOSE UNKNOWN
C
C NOTES: THIS ROUTINE IS NOT YET PROPERLY ANNOTATED
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 18-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - PUT UNDER SCCS CONTROL
C
C-----------------------------------------------------------------------
C
      INTEGER             N
      REAL*8              H(10),       W(10,10)
```

4.155 stark: Subroutine stark from library adas3xx

```
      SUBROUTINE STARK( AMDEUT , AMSS   ,
     &                  BENER  , DV1    , DV2    , DV3   , DENSB ,
     &                  BMAG   , DB1    , DB2    , DB3   ,
     &                  EMAG   , DE1    , DE2    , DE3   ,
     &                  DO1    , DO2    , DO3    ,
     &                  POLO   , POLP   ,
     &                  DENS   , TE     , ZEFF   ,
     &                  NU     , NL     , POPU   ,
     &                  NDCOMP , NCOMP  , WVCOMP , EMCOMP
     &                )


C-----------------------------------------------------------------------
C
C  ******************** FORTRAN77 SUBROUTINE: STARK ******************
C
C  PURPOSE: Code for modelling of emission from neutral hydrogen
C           in beams.
C
C  CALLING PROGRAM: ADAS305
C
C  NOTES: Developed from JETSHP.STARK.FORT(EMIS7)
C
C  STEPS: Evaluate Stark/Zeeman shifted hydrogenic energy levels and
C         evaluate dipole matrix elements.
C
C         Calculate directional positive ion impact born cross-sections
C         for Stark/Zeeman states.
C
C         Calculate populations of excited states.
C
C         Calculate polar distribution of emitted radiation for selected
C         lines and its polarisation for the charge exchange spectroscopy
C         multichord viewing lines.
C
C         Initial basis wave functions  -  n l s ml ms
C
C         Stark field is from particle motion across the magnetic induction
C         and a separate pure electric field.
C
C         General geometry specification is by direction cosines
C         dv1,dv2,dv3          : direction cosines of beam particle velocity
C         db1,db2,db3          : direction cosines of magnetic induction.
C         delec1,delec2,delec3 : direction cosines of pure electric filed
C         do1,do2,do3          : direction cosines of observation viewing line
C
C         Specific geometry
C         viewing direction defines the   -i direction
C         i-k plane is that of viewing line and beam direction
C         normal to i-k plane defines the  j direction
C         thetv= angle of beam to i direction (deg)
C         ebeam=beam speed  (kev/amu)
C         b=magnetic induction  (tesla)
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  NU      = UPPER PRINCIPAL QUANTUM NUMBER LINE
C          (I*4)  NL      = LOWER PRINCIPAL QUANTUM NUMBER LINE
C          (I*4)  POPU    = RELATIVE POPULATION OF UPPER (NU) LEVEL
```

```
C
C          (R*8)  AMDEUT  = ATOMIC MASS OF HYDROGEN IN BEAM
C          (R*8)  AMSS    = ATOMIC MASS OF HYDROGEN IN PLASMA
C          (R*8)  BENERA  = ENERGY OF ITH BEAM COMPONENT (EV/AMU)
C          (R*8)  DV1     = D.C. FOR X-CPT OF BEAM VELOCITY
C          (R*8)  DV2     = D.C. FOR Y-CPT OF BEAM VELOCITY
C          (R*8)  DV3     = D.C. FOR Z-CPT OF BEAM VELOCITY
C          (R*8)  DENSB   = SPECIFIC NEUTRAL BEAM DENSITY (CM-3)
C          (R*8)  BMAG    = SPECIFIC MAGNETIC FIELD INDUCTION (TESLA)
C          (R*8)  DB1     = D.C. FOR X-CPT OF BMAG
C          (R*8)  DB2     = D.C. FOR Y-CPT OF BMAG
C          (R*8)  DB3     = D.C. FOR Z-CPT OF BMAG
C          (R*8)  EMAG    = SPECIFIC ELECTRIC FIELD STRENGTH (VOLTS)
C          (R*8)  DE1     = D.C. FOR X-CPT OF EMAG
C          (R*8)  DE2     = D.C. FOR Y-CPT OF EMAG
C          (R*8)  DE3     = D.C. FOR Z-CPT OF EMAG
C          (R*8)  DO1     = D.C. FOR X-CPT OF SPECIFIC VIEWING LINE
C          (R*8)  DO2     = D.C. FOR Y-CPT OF SPECIFIC VIEWING LINE
C          (R*8)  DO3     = D.C. FOR Z-CPT OF SPECIFIC VIEWING LINE
C          (R*8)  POLO    = SPECIFIC SIGMA POLARISATION INTENSITY MULTIPLIER
C          (R*8)  POLP    = SPECIFIC  PI   POLARISATION INTENSITY MULTIPLIER
C          (R*8)  DENS    = SPECIFIC PLASMA ELECTRON DENSITY (CM-3)
C          (R*8)  TE      = SPECIFIC PLASMA ELECTRON TEMPERATURE (EV)
C          (R*8)  ZEFF    = SPECIFIC PLASMA EFFECTIVE Z
C
C          (L)    LPASS   = IF TRUE OUTPUT A LOG FILE
C
C
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          BORNP1     ADAS      Stage 1 Born cross-section calculation
C          BORNP2     ADAS      Stage 2
C          DIPOL      ADAS      H Dipole length radial matrix elements
C          GAMAF      ADAS      Stack vector of factorial function
C          STARK2     ADAS      Calc. Stark perturb. matrix elements
C          UNBUN2     ADAS      Extract indiv. set qu. nos. from integer
C          ZEEMN2     ADAS      Calc. Zeeman perturb. matrix elements
C          C5RLSP     ADAS      Calc. rel.+s.o. energy matrix elements
C          HYDEMI     ADAS      Collisional mixing of H excited levels
C          ZHPEV      LAPACK    Compute eigenvectors of complex
C                               Hermitian matrix
C
C
C AUTHOR:  H.P.SUMMERS, JET
C          14 SEPT 1989
C
C-----------------------------------------------------------------------
C
C
C ADAS305 version - originally SPSTRK.
C
C VERSION  : 1.1
C DATE     : 24-02-2005
C MODIFIED : Martin O'Mullane
C             - First version. Restrict to a single track.
C
C VERSION  : 1.2
C DATE     : 24-01-2006
C MODIFIED : Hugh Summers
```

```
C             - introduced relativistic +spin-orbit fine structure
C             - placed beam velocity vector direction cosines in call
C               parameters and made general
C             - corrected AMSS to AMDEUT for beam atom energy levels
C
C VERSION  : 1.3
C DATE     : 28-09-2006
C MODIFIED : Martin O'Mullane
C             - Add missing blockdata for WF, XF, ABETA and F
C               (prefixed variable with bd_ to avoid name clashes).
C             - INDW3A initialised to zero for all 726 values.
C             - Some more details sent to pass file.
C
C VERSION  : 1.4
C DATE     : 28-09-2006
C MODIFIED : Martin O'Mullane
C             - Setup infrastructure for matching upper level to adf22
C               data but don't make correction yet.
C
C-----------------------------------------------------------------------
      INTEGER             NCOMP,       NDCOMP,      NL,          NU
      REAL*8              AMDEUT,      AMSS,        BENER,       BMAG
      REAL*8              DB1,         DB2,         DB3,         DE1
      REAL*8              DE2,         DE3,         DENS,        DENSB
      REAL*8              DO1,         DO2,         DO3,         DV1
      REAL*8              DV2,         DV3,         EMAG
      REAL*8              EMCOMP(NDCOMP),           POLO,        POLP
      REAL*8              POPU,        TE,          WVCOMP(NDCOMP)
      REAL*8              ZEFF
```

4.156 start7: Subroutine start7 from library adas3xx

```
       SUBROUTINE START7(IUTMP,IUPS1,IUPS2,STITLE,DSLPATH,
     &                   NBENG,NTEMP,NDENS, lbndl, lproj)
       IMPLICIT REAL*8(A-H,O-Z)
C-----------------------------------------------------------------------
C
C       *********  FORTRAN 77 ROUTINE : START7.F **********
C
C       PURPOSE : CALCULATION OF THE RESOLVED-NL POPULATION
C                 STRUCTURE.
C
C       NOTE    : THE RELEVANT QUANTUM NUMBERS ASSOCIATED WITH
C                 THE POPULATION STRUCTURE CALCULATION ARE
C                 STORED IN INTEGER ARRAYS WERE EACH ELEMENT
C                 IS 4 BYTES. THE FIRST 10 BITS OF THE 4 BYTE
C                 INTEGER ELEMENT OF THE ARRAY IS USED TO
C                 STORE THE TOTAL ANGULAR MOMENTUM QUANTUM
C                 NUMBER,L. THE NEXT TEN BITS IS USED TO STORE
C                 THE ORBITAL ANGULAR MOMENTUM QUANTUM
C                 NUMBER,l. THE LAST 12 BITS ARE USED TO STORE
C                 THE PRINCIPAL QUANTUM NUMBER,N.
C
C                 |<---------4 BYTE INTEGER--------->|
C                 |<---------32 BIT INTEGER--------->|
C                 |<----N----->|<----l--->|<---L---->|
C
C               [ |.....N.....|.....l....|.....L....|  ]
C
C                 BIT OPERATORS ARE THEN EMPLOYED TO
C                 INTEROGATE ARRAYS,E.G IAND,ISHFR,
C                 USING HEXIDECIMAL MASKS.
C
C
C       INPUT   :
C
C       (CHR)   TITLE()     : NAME OF THE NEUTRAL BEAM SPECIES.
C       (I*4)   MN          :
C       (R*8)   RX3         :
C       (R*8)   DPT         :
C       (R*8)   EHCT        : CRITICAL ENERGY ?.
C       (I*4)   NHCT        : CRITICAL PRINCIPAL QUANTUM
C                             NUMBER ?.
C       (I*4)   LHCT        : CRITICAL ORBITAL QUANTUM
C                             NUMBER ?.
C       (I*4)   NIP         : RANGE OF DELTA N FOR IMPACT
C                             PARAMETER XSECTS. (LE.4)
C       (I*4)   NEX         :
C       (I*4)   IPRT        :
C       (I*4)   NDEL        :
C       (I*4)   INTD        : ORDER OF MAXWELL QUADRATURE
C                             FOR XSECTS. (LE.3)
C       (I*4)   IPRS        : 0 DEFAULT TO VAN REGEMORTER XSECTS.
C                               BEYOND NIP RANGE
C                             1 USE PERCIVAL-RICHARDS XSECTS.
C                               BEYOND NIP RANGE
C       (I*4)   ILOW        : 0 NO SPECIAL LOW LEVEL DATA ACCESSED.
C                             1 SPECIAL LOW LEVEL DATA ACCESSED.
C       (I*4)   IONIP       : 0 NO ION IMPACT COLLISIONS INCLUDED.
C                             1 ION IMPACT EXCITATION AND IONISATION
C                               INCLUDED.
```

```
C
C       (I*4)   NIONIP     : RANGE OF DELTA N FOR ION IMPACT
C                            EXCITATION XSECTS.
C       (I*4)   ILPRS      : 0 DEFAULT TO VAINSHTEIN ION IMPACT
C                              EXCITATION XSECTS.
C                            1 USE LODGE-PERCIVAL-RICHARDS ION
C                              IMPACT EXCITATION XSECTS.
C       (I*4)   IVDISP     : 0 ION IMPACT AT THERMAL MAXWELLIAN
C                              ENERGIES
C                            1 ION IMPACT AT DISPLACED THERMAL
C                              ENERGIES ACCORDING TO THE NEUTRAL
C                              BEAM ENERGY PARAMETER. IF(IVDISP=0 THEN
C                              SPECIAL LOW LEVEL DATA FOR ION IMPACT
C                              IS NOT SUBSTITUTED - ONLY VAINSHTEIN
C                              AND LODGE ET AL. OPTIONS ARE OPEN.
C                              ELECTRON IMPACT DATA SUBSTITUTION
C                              DOES OCCUR.
C       (R*4)   ZEFF       : EFFECTIVE CHARGE OF THE PLASMA.
C       (R*8)   DEDEG      : CRITICAL TRANSITION ENERGY (RYDBERGS)
C                            USED IN THE NEARLY DEGENERATE LEVEL
C                            TREATMENT.
C                            IF DE<=DEDEG THEN ASSUME ZERO A-VALUE
C                            AND NO SUPPLEMENTARY DATA.
C                            IF DE> DEDEG THEN ASSUME A-VALUE
C                            CALCULABLE AND SEARCH FOR SUPPLEMENTARY
C                            DATA.
C                            N.B. APPLIES TO DELTA N TRANSITIONS
C                            ONLY.
C
C       (I*4)   NL1        : PRINCIPAL QUANTUM NUMBER FROM WHICH
C                            THE RESOLVED-NL POPULATION STRUCTURE
C                            CALCULATION STARTS FROM
C
C       (I*4)   NL2        : PRINCIPAL QUANTUM NUMBER WHICH MARKS
C                            THE END OF THE RESOLVED-NL TREATMENT
C                            WITHIN THE POPULATION STRUCTURE
C                            CALCULATION AND INDICATES THE START
C                            OF THE BUNDLED-N APPROXIMATION.
C       (I*4)   NL3        : UPPER PRINCIPAL QUANTUM NUMBER OF
C                            THE BUNDLED-N APPROXIMATION.
C       (R*8)   Z0         : NUCLEAR CHARGE OF BEAM ATOM ?.
C       (R*8)   Z1         : ION CHARGE+1 OF BEAM ION ?.
C       (R*8)   ALF        : ADJUSTABLE PARAMETER ASSOCIATED
C                            WITH THE MODIFIED POTENTIAL USED
C                            WHEN SOLVING THE RADIAL WAVE
C                            EQUATION.
C       (R*8)   AMSZ0      :
C       (R*8)   AMSHYD     :
C       (I*4)   LP         :
C       (I*4)   ISP        :
C
C
C       OUTPUT  :
C
C               (R*8)  ....ETC
C
C       GENERAL :
C
C       (I*4)   NLREP()    : ARRAY CONTAINING REPRESENTATIVE
C                            LEVELS.
C       (R*8)   ENL()      : EFFECTIVE PRINCIPAL QUANTUM
```

```
C                             NUMBER.
C       (R*8)   ENL2()      : RECIPROCAL OF THE EFFECTIVE PRINCIPAL
C                             QUANTUM NUMBER SQUARED.
C       (I*4)   KPF()       : ARRAY CONTAINING THE QUANTUM NUMBERS,
C                             N,l,L FROM NMIN TO NMAX, IN ORDER
C                             OF DECREASING BINDING ENERGY. SEE
C                             NOTE AT THE TOP OF PROGRAM.
C       (I*4)   KPB()       : ARRAY CONTAINING THE INDEX OF THE
C                             CORRESPONDING LEVEL IN KPF().
C       (R*8)   EGY         : IONISATION POTENTIAL ( RYDBERGS ).
C       (I*4)   IR          : COUNTER TO REFERENCE REPRESENTATIVE
C                             LEVELS.
C       (I*4)   I           : GENERAL COUNTER.
C       (R*8)   V           : EFFECTIVE PRINCIPAL QUANTUM NUMBER.
C       (R*8)   E           : RECIPROCAL OF THE EFFECTIVE PRINCIPAL
C                             QUANTUM NUMBER SQUARED.
C       (R*8)   EXE         : VARIABLE USED TO ASSIGN THE VALUE
C                             OF EXP(I/k*Te).
C       (R*8)   EXS         : VARIABLE USED TO ASSIGN THE VALUE
C                             OF EXP(I/k*TS)
C       (I*4)   K           : GENERAL COUNTER.
C       (R*8)   C1()        : COEFFICIENT OF THE QUANTUM DEFECT
C                             EXPANSION.
C       (R*8)   C2()        : COEFFICIENT OF THE QUANTUM DEFECT
C                             EXPANSION.
C       (R*8)   C3()        : COEFFICIENT OF THE QUANTUM DEFECT
C                             EXPANSION.
C
C
C
C
C
C       ROUTINES:
C
C               ROUTINE        SOURCE     BRIEF DESCRIPTION
C               -----------------------------------------------
c               SUPPHE1         ADAS    OBTAINS FUNDAMENTAL DATA
C                                       FROM APPROPRIATE DATABASES.
C               OVLP            ADAS    ??????????????????????????
C               SETUP3          ADAS    ??????????????????????????
C               SPIJ            ADAS    ??????????????????????????
C               CCNST7          ADAS    ASSEMBLES ARRAYS USED TO
C                                       CONSTRUCT THE COLLISIONAL-
C                                       RADIATIVE MATRIX.
C               CCNSE4          ADAS    APPLIES MATRIX CONDENSATION
C                                       SCHEME TO ARRAYS USED TO
C                                       ASSEMBLE THE COLLISIONAL-
C                                       RADIATIVE MATRIX.
C               HYSCL           ADAS    ??????????????????????????
C
C
C       HISTORY : ORIGINALLY WRITTEN BY H.P.SUMMERS.
C
C       NOTE    : THE RESOLVED-NL CALCULATION WAS STRUCTURED
C                 IN SUCH A MANNER THAT THE CALCULATION WAS
C                 PERFORMED IN TWO STAGES. THIS TWO STAGE
C                 PROCCESS HAS BEEN REMOVED. CUBIC SPLINE
C                 INTERPLOATION IN L. THE DIMENSIONALITIES
C                 FOR EACH SPIN SYSTEM ARE AS FOLLOWS :
C
C                 NUMBER OF LEVELS <1000.
```

```
C                 NUMBER OF PRINCIPAL QUANTUM LEVELS<300.
C                 NUMBER OF RESOLVED PRINCIPAL QUANTUM LEVELS<40.
C                 NUMBER OF RESOLVED LEVELS <800
C                 NUMBER OF RESOLVED REPR.PRINC.QUANTUM LEVELS<11.
C                 NUMBER OF REPRESENTATIVE LEVELS<80.
C                 NUMBER OF PRINCIPAL QUANTUM REPRESENTATIVE LEVELS<30.
C
C
C       CONTACT : HARVEY ANDERSON
C                 UNIVERSITY OF STRATHCLYDE
C                 ANDERSON@PHYS.STRATH.AC.UK
C
C       DATE    : 2/2/98
C
C       NOTE :
C                 THE C-R MATRIX IS PASSED TO THE ROUTINE FINISH5.F
C                 VIA A SCRTACH FILE ON STREAM 12. IDEALLY THE VARIBLES
C                 SHOULD BE PASSED DIRECTLY TO THE ROUTINE.
C
C
C       MODIFIED
C
C VERSION:      1.2                                     DATE:   21-10-99
C MODIFIED: RICHARD MARTIN
C               CHANGED HEXADECIMAL CONSTANTS TO Z'FFF00000' FORM.
C
C VERSION : 1.3
C MODIFIED: Martin O'Mullane
C DATE    : 08-11-2004
C           Alter nmax in gamaf() from 200 to 500.
C
C VERSION : 1.4
C DATE    : 18-11-2004
C MODIFIED: Martin O'Mullane
C            - Align with Harvey Anderson's last version.
C            - Add lproj if projection output is requested.
C            - Add lbndl if adf36 output file is requested.
C            - The dsnps1 variable is replaced by iups2 in the
C              parameter list.
C
C VERSION : 1.5
C DATE    : 16-05-07
C MODIFIED: Allan Whiteford
C            - Moved parameter statement to below comment block
C              as part of subroutine documentation procedure.
C
C-----------------------------------------------------------------------
      CHARACTER*80        DSLPATH,     STITLE
      INTEGER             IUPS1,       IUPS2,       IUTMP,       NBENG
      INTEGER             NDENS,       NTEMP
      LOGICAL             LBNDL,       LPROJ
```

4.157 supphe1: Subroutine supphe1 from library adas3xx

```
       SUBROUTINE SUPPHE1(TEV,EBEAM,TIEV,NIMP,ZIMPA,FRIMPA, AMIMPA,
     &                    DSLPATH )
      IMPLICIT REAL*8(A-H,O-Z)
C-----------------------------------------------------------------------
C
C       ******  FORTRAN 77 ROUTINE : SUPPHE1.F  ******
C
C       PURPOSE : ACCESS FUNDAMENTAL CROSS SECTION DATA FOR THE
C                 BUNDLED NL CALCULATION.
C
C       INPUT   :
C
C       (R*8)   TEV            : ELECTRON TEMPERATURE (eV)
C       (R*8)   EBEAM          : NEUTRAL BEAM ENERGY (eV/AMU)
C       (R*8)   TIEV           : ION TEMPERATURE (eV)
C       (R*8)   ZIMP           : Z OF EFFECTIVE IMPURITY FOR ION
C                                COLLISIONS ( EXC H+ ).
C       (I*4)   ITYP1          : 1  OBTAIN ELECTRON IMPACT EXCITATION
C                                   DATA FROM SPECIFIC ION FILE.
C                                0  DO NOT OBTAIN ELECTRON IMPACT
C                                   EXCITATION DATA FROM SPECIFIC ION.
C       (I*4)   ITYP2          : 1  OBTAIN ELECTRON IMPACT IONISATION
C                                   DATA FROM SZD TYPE FILE.
C                                0  DO NOT OBTAIN ELECTRON IMPACT
C                                   IONISATION DATA FROM SZD FILE.
C       (I*4)   ITYP3          : 1  OBTAIN H+ IMPACT EXCITATION
C                                   DATA FROM ADF02 TYPE FILE.
C                                0  DO NOT OBTAIN H+ IMPACT EXCITATION
C                                   DATA FROM ADF02 TYPE FILE.
C       (I*4)   ITYP4          : 1  OBTAIN H+ IMPACT IONISATION AND
C                                   CHARGE EXCHANGE FROM ADF02 TYPE FILE.
C                                0  DO NOT OBTAIN H+ IMPACT IONISATION
C                                   AND CHARGE EXCHANGE FROM ADF02 TYPE FILE.
C       (I*4)   ITYP5          : 1  OBTAIN ZIMP ION IMPACT EXCITATION FROM
C                                   ADF02 TYPE FILE.
C                                0  DO NOT OBTAIN ZIMP ION IMPACT EXCITATION
C                                   FROM ADF02 TYPE FILE.
C       (I*4)   ITYP6          : 1  OBTAIN ZIMP ION IMPACT IONISATION AND
C                                   CHARGE EXCHANGE FROM ADF02 TYPE FILE.
C                                0  DO NOT OBTAIN ZIMP ION IMPACT IONISATION
C                                   AND CHARGE EXCHANGE FROM ADF02 TYPE FILE.
C       (CHR)   DSLPATH        : CHARCTER STRING CONTAINING THE USER NAME.
C                                INFORMATION REQUIRED TO OPEN UP LOW LEVEL
C                                DATA FILES (OBTAINED FROM IDL).
C       (C*120) TITLX          : CHARACTER STRING SPECIFYING THE SOURCE
C                                OF IONISATION DATA
C
C       OUTPUT  :
C
C       (I*4)   NSYS           : NUMBER OF SPIN SYSTEMS (=2)
C       (I*4)   ISYSA(IS)      : MULTIPLICITY OF SPIN SYSTEM
C       (I*4)   NNA(IR)        : N-SHELL     FOR COPDAT FILE LEVEL INDEX IR
C       (I*4)   ISA(IR)        : MULTIPLICITY
C       (I*4)   ILA(IR)        : TOTAL ORBITAL ANGULAR MOMENTUM
C       (R*8)   WTA(IR)        : STATISTICAL WEIGHT
C       (R*8)   ATBE(IR,IR'')  : EINSTEIN A-COEFFICIENT
C       (R*8)   XTBE(IR,IR'')  : ELECTRON IMPACT EXCITATION RATE COEFFICIENT.
C       (I*4)   LXTBE(IR,IR'') : ELECTRON IMPACT EXCITATION TYPE MARKER
C                                (0 =NO VALUE, 1=VALUE)
```

```
C       (R*8)   XTBP(I,I'',IS) : H+ IMPACT EXCITATION RATE COEFFICIENT.
C       (R*8)   XTBZ(I,I'',IS) : ZIMP ION IMPACT EXCITATION RATE COEFFICIENT
C       (R*8)   STBE(I,IS)     : ELECTRON IMPACT IONISATION RATE COEFFICIENT
C       (R*8)   STBP(I,IS)     : H+ ION IMPACT IONISATION AND CHARGE
C                                EXCHANGE RATE COEFFICIENT.
C       (R*8)   STBZ(I,IS)     : ZIMP ION IMPACT IONSATION AND CHARGE
C                                EXCHANGE RATE COEFFICIENT
C       (I*4)   LXTBP(I,I'',IS): H+ IMPACT EXCITATION TYPE  MARKER
C                                (0 =NO VALUE, 1=VALUE)
C       (I*4)   LXTBZ(I,I'',IS): ZIMP ION IMPACT EXCITATION TYPE MARKER
C       (I*4)   LSTBE(I,IS)    : ELECTRON IMPACT IONISATION TYPE MARKER
C       (I*4)   LSTBP(I,IS)    : H+ IMPACT IONISATION & CHARGE EXCHANGE
C                                TYPE MARKER
C       (I*4)   LSTBZ(I,IS)    : ZIMP ION IMPACT IONISATION AND CHARGE EXCHANGE
C                                TYPE MARKER
C       (R*8)   PXTBP(I,IS)    : H+ IMPACT EXCITATION TYPE PROJECTION MULTIPLIER
C       (R*8)   PXTBZ(I,IS)    : ZIMP ION IMPACT EXCITATION TYPE PROJECTION
C                                MULTIPLIER
C       (R*8)   PSTBE(IS)      : ELECTRON IMPACT IONISATION TYPE PROJECTION
C                                MULTIPLIER
C       (R*8)   PSTBP(IS)      : H+ IONISATION & CHARGE EXCHANGE TYPE
C                                PROJECTION MULTIPLIER
C       (R*8)   PSTBZ(IS)      : ZIMP ION IMPACT IONISATION AND CHARGE EXCHANGE
C                                TYPE PROJECTION MULTIPLIER
C       (I*4)   LPXTBP(I,IS)   : H+ IMPACT EXCITAION TYPE PROJECTION MULTIPLIER
C                                USED ABOVE THIS N'
C       (I*4)   LPXTBZ(I,IS)   : ZIMP ION IMPACT EXCITATION TYPE 5 PROJECTION
C                                MULTIPLIER USED ABOVE THIS N'
C       (I*4)   LPSTBP(IS)     : H+ IMPACT IONISATION & CHARGE EXCHANGE TYPE
C                                PROJECTION MULTIPLIER USED ABOVE THIS N
C       (I*4)   LPSTBZ (IS)    : ZIMP ION IMPACT IONISATION AND CHARGE EXCHANGE
C                                TYPE PROJECTION MULTIPLIER USED ABOBE THIS N
C
C       ROUTINES:
C
C               ROUTINE         SOURCE          DESCRIPTION
C            ---------------------------------------------
C               XXSLEN          ADAS            IDENTIFY THE FIRST & LAST
C                                               NON-BLANK CHARACTER IN A
C                                               STRING.
C               XXWORD          ADAS            MANIPULATES STRINGS.
C               SZD             ADAS            RETURNS ELECTRON IMPACT
C                                               IONISATION RATES WHICH
C                                               ARE OBTAINED FROM ADF07
C                                               TYPE FILE.
C               QHE             ADAS            RETURNS BEAM/THERMAL MAXWELL
C                                               AVERAGED RATE COEFFICIENTS.
C
C
C       CONTACT : HARVEY ANDERSON
C                 UNIVERSITY OF STRATHCLYDE
C                 ANDERSON@PHYS.STRATH.AC.UK
C
C       DATE    : 26/11/97
C
C       MODIFICATIONS   :  REPLACED NAG ROUTINES WITH THE NEAR
C                          ADAS EQUIVALENT ROUTINES.
C                          HARVEY ANDERSON
C       DATE            :  10/3/99
C
C VERSION: 1.2                          DATE: 15-10-99
```

```
C MODIFIED: RICHARD MARTIN
C               REMOVED 'ACTION' FROM OPEN STATEMENT.
C
C VERSION: 1.3                          DATE: 07-07-2004
C MODIFIED: ALLAN WHITEFORD
C               -CHANGED CALLS FROM DXNB{A,B}F TO XXNB{A,B}F
C
C VERSION: 1.4                          DATE: 04-11-2004
C MODIFIED: ALLAN WHITEFORD
C               -ADDED DECLARATION OF TITLX VARIABLE
C
C VERSION : 1.5
C DATE    : 22-02-2005
C MODIFIED: Martin O'Mullane
C            - Declare ltrng as a logical.
C            - Replace TITLF with DSLPATH in calls to qhe for
C              itypes 4 and 5.
C
C VERSION : 1.6
C DATE    : 16-05-07
C MODIFIED: Allan Whiteford
C            - Moved parameter statement to below comment block
C              as part of subroutine documentation procedure.
C
C-----------------------------------------------------------------------
      CHARACTER*80        DSLPATH
      INTEGER             NIMP
      REAL*8              AMIMPA(10),  EBEAM,       FRIMPA(10),  TEV
      REAL*8              TIEV,        ZIMPA(10)
```

4.158 v2bnmod: Subroutine v2bnmod from library adas3xx

```
       SUBROUTINE V2BNMOD(IPOSNT , JDENS  , JTE    , NREP   , F1     ,
     &                    F2     , F3     , BNCALC , BNACT  , XPOP   ,
     &                    IMAX   , DENSH  , DENS   , DENSP  , TE     ,
     &                    TP     , BMENER , FLUX   , DEXPTE , ALFA   ,
     &                    S      , DSLPATH, NIMP   , ZIMPA  , ZEFF   ,
     &                    DNIMPA)

       IMPLICIT REAL*8 (A-H,O-Z)

C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: V2BNMOD *******************
C
C  VERSION:  (ADAS91) - SEE SC CS COMMENTS FOR VERSION NO.
C
C  PURPOSE:  THIS CODE PERFORMS THE ACTUAL CALCULATIONS FOR ADAS 310
C            IT IS IN AN INCOMPLETE VERSION AND THESE COMMENTS DO NOT
C            YET FOLLOW THE STANDARD ADAS PATTERN.
C-----------------------------------------------------------------------
C      CALCULATION OF BN -1 CASE A,B
C
C      EXCIT. XSECT. OPTIONS:
C      ----------------------
C                     (A) VAN REGEMORTER    - ELECTRONS
C                     (B) IMPACT PARAMETER  - ELECTRONS
C                                             PROTONS
C                     (C) PERCIVAL-RICHARDS - ELECTRONS
C                                           - PROTONS & ZIMP IONS
C                     (D) VAINSHTEIN        - PROTONS & ZIMP IONS
C                     (E) SPECIAL LOW LEVEL - ELECTRONS
C                                           - PROTONS & ZIMP IONS
C      IONIS. XSECT. OPTIONS:
C      ----------------------
C                     (A) ECIP              - ELECTRONS
C                     (B) PERCIVAL-RICHARDS - PROTONS & ZIMP IONS
C                     (C) SPECIAL LOW LEVEL - ELECTRONS
C                                           - PROTONS & ZIMP IONS
C      CX RECOM. XSECT. OPTIONS:
C      -------------------------
C                     (A) SPECIAL           - H(1S) DONOR
C
C
C  NOTES
C  -----
C      (A) SPECIAL LOW LEVEL DATA ACCESSED BY SPECIFIC ION ROUTINE
C
C          ION   ROUTINE    ACCESSED FILE              ACQUIS. ROUTINE
C          ---   -------    -------------              ---------------
C           H0    NSUPH1    IONATOM.DATA(H)                 QH
C                           HLIKE.DATA(AGG1984)              -
C
C
C      (B) SPECIAL CHARGE EXCHANGE DATA FROM  CHEXDATA.DATA
C
C          ION    DONOR     ROUTINE    DATA MEMBER     ACQUIS. ROUTINE
C          ---    -----     -------    -----------     ---------------
C           H0    H(1S)     BNQCTB       H1NEW1            BNQCTB
C
C
```

```
C
C
C  INPUT
C      IPOSNT   = 0  BYPASS
C               = 1  LEAVE W1.NE.0 AND FORCE DENSH.EQ.0
C               = 2  FORCE W1.EQ.0 AND FORCE DENSH.EQ.0
C               = 3  LEAVE W1.NE.0 AND LEAVE DENSH.NE.0
C               = 4  FORCE W1.EQ.0 AND LEAVE DENSH.NE.0
C               = 5  BYPASS
C      JDENS    = DENSITY     SELECTOR
C      JTE      = TEMPERATURE SELECTOR
C      NREP(I)  = PRINCIPAL QUANTUM NUMBER OF ITH REPRESENTATIVE LEVEL
C      XPOP(I)  = POPULATIONS OF REPRESENTATIVE LEVELS
C      IMAX     = NUMBER OF REPRESENTATIVE LEVELS
C      DENSH    = NEUTRAL HYDROGEN DENSITY IN BEAM  (CM-3)
C      DENS     = ELECTRON DENSITY  (CM-3)
C      DENSP    = PROTON/DEUTERON DENSITY  (CM-3)
C      TE       = ELECTRON TEMPERATURE (K)
C      TP       = PROTON/DEUTERON TEMPERATURE (K)   (SAME FOR ZIMP IONS)
C      BMENER   = NEUTRAL BEAM PARTICLE ENERGY  (EV/AMU)
C      FLUX     = NEUTRAL BEAM FLUX  (CM-2 SEC-1)
C      DEXPTE(I)= EXP(ATE/NREP(I)**2) FOR ITHE REPRESENTATIVE LEVEL
C      DSLPATH  = STRING CONTAINING PATH TO THE REQUIRED INPUT FILES
C                 CALLED BY NSUPH1.FOR
C
C  OUTPUT
C      F1       =
C      F2       =
C      F3       =
C      BNCALC   =
C      BNACT    =
C      ALPHA    =
C      S        =
C
C  INPUT SPECIFICATION FOR STREAM 51 DATA FILE
C
C      NUCCHG = NUCLEAR CHARGE
C      EXMEMB = DATA SET NAME OF EXPANSION FILE USED BY CLDLBN2
C      CXMEMB = DATA SET NAME FOR CHARGE EXCHANGE DATA TO BE USED BY
C               BNQCTB
C      IBLOCK = SUB-BLOCK SELECTOR FOR CXMEMB
C
C      JDENSM = NUMBER OF DENSITIES
C      JTEM   = NUMBER OF TEMPERATURES
C      TS     = EXTERNAL RADIATION FIELD TEMPERATURE (K)
C      W      = EXTERNAL RADIATION FIELD DILUTION FACTOR (HIGH LEVELS)
C      Z      = RECOMBINING ION CHARGE  (Z1 IN USUAL NOTATION)
C      CION   = MULTIPLIER OF GROUND LEVEL ELECTRON IMPACT IONISATION
C               RATE COEFFICIENT
C      CPY    = MULTIPIER ON ELECTRON IMPACT EXCITATION RATE COEFFICIENT
C               FORM THE GROUND LEVEL
C      W1     = EXTERNAL RADIATION FIELD DILUTION FACTOR FOR PHOTO-IONI
C               SATION FORM THE GROUND LEVEL.
C
C      NIP    = RANGE OF DELTA N FOR IMPACT PARAMETER XSECTS. (LE.4)
C      INTD   = ORDER OF MAXWELL QUADRATURE FOR XSECTS. (LE.3)
C      IPRS   = 0   DEFAULT TO VAN REGEMORTER XSECTS. BEYOND NIP RANGE
C               1   USE PERCIVAL-RICHARDS XSECTS.     BEYOND NIP RANGE
C      ILOW   = 0   NO SPECIAL LOW LEVEL DATA ACCESSED
C               1      SPECIAL LOW LEVEL DATA ACCESSED
C      IONIP  = 0   NO ION IMPACT COLLISIONS INCLUDED
```

```
C               1      ION IMPACT EXCITATION AND IONISATION INCLUDED
C      NIONIP = RANGE OF DELTA N FOR ION IMPACT EXCITATION XSECTS.
C      ILPRS  = 0   DEFAULT TO VAINSHTEIN XSECTS.
C               1   USE LODGE-PERCIVAL-RICHARDS XSECTS.
C      IVDISP = 0   ION IMPACT AT THERMAL MAXWELLIAN ENERGIES
C               1   ION IMPACT AT DISPLACED THERMAL ENERGIES ACCORDING
C                   TO THE NEUTRAL BEAM ENERGY PARAMETER
C                   *   IF(IVDISP=0 THEN SPECIAL LOW LEVEL DATA FOR ION
C                       IMPACT IS NOT SUBSTITUTED - ONLY VAINSHTEIN AND
C                       LODGE ET AL. OPTIONS ARE OPEN.  ELECTRON IMPACT
C                       DATA SUBSTITUTION DOES OCCUR.
C      1   = PLASMA Z EFFECTIVE
C      NOSCAN = 0   EXECUTE SCANNING VERSION OF CODE
C               1   EXECUTE SIMULTANEOUS IMPURITY NO SCAN RUN
C      NIMP   = NUMBER OF IMPURITIES (EXCL.H+) IN NO SCAN CASE
C  ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          MATINV     ADAS      MATRIX INVERSION WITH ACCOMPANYING
C                               SOLUTION OF LINEAR EQUATIONS
C          DIEL_310    ????
C
C
C
C  ********* H.P.SUMMERS, JET               11 APR 1990  ***************
C  *********                          ALT. 17 JUL 1991  SUB. RQINEW,
C                                                       RQLNEW, RQVNEW
C  *********                          ALT. 10 JAN 1994  ALLOW MULTIPLE
C                                                       SIMULT.IMPURITY
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
C
C UPDATE:  19/01/94 - JONATHAN NASH - TESSELLA SUPPORT SERVICES PLC
C
C          THE FOLLOWING MODIFICATIONS HAVE BEEN MADE TO THE SUBROUTINE:
C
C          1) THE INPUT FILE UNIT NUMBER HAS BEEN CHANGED FROM 5 TO 51.
C
C          2) THE SIZES OF 'EXMEMB' AND 'CXMEMB' HAVE BEEN INCREASED
C             AS THEY NOW CONTAIN THE FULL DATA SET NAMES RATHER THAN
C             JUST THE MEMBER NAMES.
C
C          3) A PARAMETER FLAG HAS BEEN ADDED TO SWITCH ON/OFF
C             DIAGNOSTIC PRINTING (UNIT 6).
C
C  NOTES:  NO ATTEMPT HAS BEEN MADE TO RESTRUCTURE THE ROUTINE. RATHER
C          THE MINIMUM AMOUNT OF WORK TO INTEGRATE THE ROUTINE INTO
C          ADAS310 HAS BEEN COMPLETED.
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 12-12-95
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST VERSION
C               - REPLACED ALL HOLLERITH CONSTANTS H0 ANF H1 WITH
C                 H<SPACE>
C               - REPLACED HOLLERITH CONSTANTS WITH STANDARD STRINGS
C                 IN FORMAT STATEMENTS 117, 128 AND 129.
C               - TIDIED UP PARTS OF THE COMMENTS AND CODE
C               - ADDED STRING DSLPATH TO BE USED TO CONSTRUCT UNIX
```

```
C                 STYLE FILENAMES IN NSUPH1.FOR
C
C VERSION: 1.2                          DATE: 24-01-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - ADDED NBENG TO INPUT LIST AND CALL TO BNQCTB
C
C VERSION: 1.3                          DATE: 24-01-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - REVERSED ABOVE CHANGE
C
C VERSION: 1.4                          DATE: 27-09-96
C MODIFIED: WILLIAM OSBORN + HUGH SUMMERS
C               - REMOVED MISTAKEN TEST FOR 1.LE.0.0D0 IN
C                 THE CASE OF MULTIPLE IMPURITIES.
C VERSION: 1.5
C MODIFIED: HARVEY ANDERSON
C               - IMPROVED THE HANDLING OF MULTIPLE IMPURITIES TO
C                 INCLUDE HYDROGEN.
C               - ALTERED THE CALLING STRUCTURE OF THIS ROUTINE TO
C                 ALLOW ADDITIONAL VARIABLES TO BE PASSED TO RUN310.
C               - INCREASED THE SIZE OF THE ARRAYS DENSA, DENPA, TEA
C                 AND TPA FROM 10 TO 25.
C
C VERSION : 1.6
C DATE    : 04-04-2000
C MODIFIED: RICHARD MARTIN
C               - CHANGED NAME OF DIEL SUBROUTINE TO DIEL_310
C
C VERSION : 1.7
C DATE    : 22-02-2005
C MODIFIED: Martin O'Mullane
C               - A real*8 parameter of cldlbn2 which was passed as an
C                 integer (1) to cldlbn2 is changed to a double (1.0D0).
C               - Direct debug output stream to unit 0.
C               - Remove IBM error underflow and errset routines.
C
C VERSION : 1.8
C DATE    : 27-06-2007
C MODIFIED: Martin O'Mullane
C               - Add lpass as argument to cldlbn2. Set to .TRUE.
C
C-----------------------------------------------------------------------
      CHARACTER*80        DSLPATH
      INTEGER             IMAX,         IPOSNT,      JDENS,       JTE
      INTEGER             NIMP,         NREP(31)
      REAL*8              ALFA,         BMENER,      BNACT(30)
      REAL*8              BNCALC(30),   DENS,        DENSH,       DENSP
      REAL*8              DEXPTE(550), DNIMPA(10),  F1(30),      F2(30)
      REAL*8              F3(30),       FLUX,        S,           TE
      REAL*8              TP,           XPOP(2),     ZEFF
      REAL*8              ZIMPA(10)
```

4.159 xip: Subroutine xip from library adas3xx

```
       FUNCTION XIP(XI,DELTA)
C
       IMPLICIT REAL*8 (A-H,O-Z)
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 FUNCTION: XIP *************************
C
C PURPOSE UNKNOWN
C
C NOTES: THIS ROUTINE IS NOT YET PROPERLY ANNOTATED
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 16-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST VERSION
C
C-----------------------------------------------------------------------
C
       XM=DABS(XI)
       X1=3.142*(DELTA+0.25+0.03125/(DELTA+0.5))*DEXP(-2.0*DELTA
     1 +1.1416*(XM+XM*XM)/(1.0+0.6*XM+3.0*XM*XM/(6.0+2.0*DELTA)))
       X2=3.0/(XM+3.0)
       X3=0.2/(XM+0.4)
       XIP=X1*(1.0-X2+X2*X2+X3-2.0*X3*X3)
       IF(XI)1,2,2
    1  XIP=XIP*DEXP(6.283*XI)
    2  CONTINUE
       RETURN
       END
      REAL*8              DELTA,       XI
```

4.160 yip: Subroutine yip from library adas3xx

```
       FUNCTION YIP(XI,DELTA)
C
       IMPLICIT REAL*8 (A-H,O-Z)
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 FUNCTION: YIP *************************
C
C PURPOSE UNKNOWN
C
C NOTES: THIS ROUTINE IS NOT YET PROPERLY ANNOTATED
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 16-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST VERSION
C
C-----------------------------------------------------------------------
C
      REAL*8              DELTA,       XI
```

4.161 zero1: Subroutine zero1 from library adas3xx

```
       SUBROUTINE ZERO1(A,B,VA,VB,D1,X,XI,Z,E,TIF,T2)
C
       IMPLICIT REAL*8 (A-H,O-Z)
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: ZERO1 *********************
C
C PURPOSE UNKNOWN
C
C NOTES: THIS ROUTINE IS NOT YET PROPERLY ANNOTATED
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 16-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST VERSION
C
C-----------------------------------------------------------------------
C
       X=(A*VB-B*VA)/(VB-VA)
    9  X1=X
       T=Z+E*X
       D=TIF*X
       V=0.5-T2*XIP(XI,D)/(T*T)
       IF(V)11,16,10
   10     B=X
       VB=V
       GO TO 12
   11     A=X
       VA=V
   12     X=0.5*(A+B)
       T=Z+E*X
       D=TIF*X
       V=0.5-T2*XIP(XI,D)/(T*T)
       IF(V)14,16,13
   13     B=X
       VB=V
       GO TO 15
   14     A=X
       VA=V
   15     X=(A*VB-B*VA)/(VB-VA)
       IF(DABS(X1-X)/X-D1)16,16,9
   16     RETURN
      END
      REAL*8              A,           B,           D1,          E
      REAL*8              T2,          TIF,         VA,          VB
      REAL*8              X,           XI,          Z
```

5 Subroutine library adas4xx

5.1 bnd404a: Subroutine bnd404a from library adas4xx

```
       SUBROUTINE BND404A(ITYPE  ,
     &                    NUTMAX , NUDMAX , NUZMAX , NUMMAX ,
     &                    MAXT   , MAXD   ,
     &                    IZL    , IZH    , IZ0    ,
     &                    TEK    , DENSA  ,
     &                    METFRC ,
     &                    NGRD   ,
     &                    IST2   , IST5   , IWRITE , DATE,
     &                    DSNIN)
C
C-----------------------------------------------------------------------
C
C ************ FORTRAN 77 SUBROUTINE BND404A ***************************
C
C   VERSION 1.0
C
C   PURPOSE:
C           TO FETCH DATA FROM RESOLVED ADF10 FILES, SPLINE THEM
C           ONTO THE REQUESTED TEMPERATURE/DENSITY GRID, BUNDLE
C           THEM INTO UNRESOLVED DATA USING THE INPUT METASTABLE
C           FRACTIONS, AND WRITE THE RESULT TO ADF11 FILES.
C
C   CALLING ROUTINE / PROGRAM : LH404RU / ADAS404
C
C   DATA:
C
C           THE SOURCE DATA IS CONTAINED AS MEMBERS OF PARTITIONED
C           DATA SETS AS FOLLOWS:
C
C             1. JETUID.ACD<YR>.DATA
C             2. JETUID.SCD<YR>.DATA
C             3. JETUID.CCD<YR>.DATA
C             4. JETUID.PRB<YR>.DATA
C             5. JETUID.PRC<YR>.DATA
C
C           WHERE <YR> DENOTES TWO INTEGERS FOR THE YEAR SELECTED.
C
C   SUBROUTINE:
C
C   INPUT : (I*4)  ITYPE  - TYPE OF ADF10 DATA BEING READ (SEE ABOVE)
C   INPUT : (I*4)  NUTMAX - OUTPUT ELEMENT MASTER FILE
C                           MAXIMUM NUMBER OF TEMPERATURES
C   INPUT : (I*4)  NUDMAX - OUTPUT ELEMENT MASTER FILE
C                           MAXIMUM NUMBER OF DENSITIES
C   INPUT : (I*4)  NUZMAX - OUTPUT ELEMENT MASTER FILE
C                           MAXIMUM NUMBER OF CHARGE STATES
C   INPUT : (I*4)  NUMMAX - OUTPUT ELEMENT MASTER FILE
C                           MAXIMUM NUMBER OF METASTABLES
C   INPUT : (I*4)  MAXT   - OUTPUT ELEMENT MASTER FILE
C                           ACTUAL NUMBER OF TEMPERATURES
C   INPUT : (I*4)  MAXD   - OUTPUT ELEMENT MASTER FILE
C                           ACTUAL NUMBER OF DENSITIES
C   INPUT : (I*4)  IZL    - LOWEST ION CHARGE TO READ
C   INPUT : (I*4)  IZH    - HIGHEST ION CHARGE TO READ
C   INPUT : (I*4)  IZ0    - NUCLEAR CHARGE TO READ
C   INPUT : (R*8)  DENSA()- OUTPUT ELEMENT MASTER FILE
C                           SET OF MAXD DENSITIES
```

```
C   INPUT : (R*8)  TEK()  - OUTPUT ELEMENT MASTER FILE
C                           SET OF MAXT TEMPERATURES
C   INPUT : (R*8)  METFRC(,,,) - METASTABLE POPULATION FRACTIONS,
C                           SPLINED ONTO THE OUTPUT TEMPERATURES
C                           AND DENSITIES
C                           1ST DIMENSION - DENSITY INDEX
C                           2ND DIMENSION - TEMPERATURE INDEX
C                           3RD DIMENSION - CHARGE STATE INDEX
C                           4TH DIMENSION - METASTABLE INDEX
C   INPUT : (I*4)  NGRD() - NUMBER OF GROUND STATES OF THE FIRST
C                           50 ISOELECTRONIC SEQUENCES
C   INPUT : (I*4)  IST2   - UNIT NUMBER FOR OUTPUT INFORMATION
C                           AND ERROR MESSAGES
C   INPUT : (I*4)  IST5   - UNIT NUMBER FOR READING MASTER CONDENSED
C                           FILE
C   INPUT : (I*4)  IWRITE - UNIT NUMBER FOR WRITING ADF11 DATA
C   INPUT : (C*8)  DATE   - CURRENT DATE
C
C   PARAMETER : (I*4)  NTDMAX - SIZE OF LOCAL WORKING SPACE
C                           (MUST BE GREATER THAN NUTMAX & NUDMAX)
C   PARAMETER : (I*4)  NDZ1V  - MASTER CONDENSED FILE
C                           MAXIMUM NUMBER OF CHARGE STATES
C   PARAMETER : (I*4)  NDTIN  - MASTER CONDENSED FILE
C                           MAXIMUM NUMBER OF TEMPERATURES
C   PARAMETER : (I*4)  NDDEN  - MASTER CONDENSED FILE
C                           MAXIMUM NUMBER OF DENSITIES
C
C         : (R*8)  DENSR()    - INPUT MASTER CONDENSED FILE
C                           SET OF IDE REDUCED DENSITIES
C         : (R*8)  TR()       - INPUT MASTER CONDENSED FILE
C                           SET OF ITE REDUCED TEMPERATURES
C         : (R*8)  ZIPT()     - INPUT MASTER CONDENSED FILE
C                           SET OF IZE RECOMBINING ION CHARGES
C         : (R*8)  AIPT(,,)   - INPUT MASTER CONDENSED FILE
C                           RELEVANT RATE COEFFICIENTS
C                           1ST DIMENSION - DENSITY INDEX
C                           2ND DIMENSION - TEMPERATURE INDEX
C                           3RD DIMENSION - CHARGE STATE INDEX
C         : (R*8)  EIA()      - INPUT MASTER CONDENSED FILE
C                           SET OF IONISATION POTENTIALS (CM-1)
C
C         : (R*8)  ATTY(,)    - WORK SPACE FOR INTERPOLATION
C                           - STORES LOG10(INTERPOLATED VALUES)
C                           1ST DIMENSION - TEMPERATURE
C                           2ND DIMENSION - DENSITY
C         : (R*8)  ARRAY(,)   - STORES LOG10(INTERPOLATED VALUES)
C                           1ST DIMENSION - TEMPERATURE
C                           2ND DIMENSION - DENSITY
C         : (R*8)  WORK(,)    - SUM OF INTERPOLATED VALUES OVER
C                           METASTABLE STATES
C                           1ST DIMENSION - TEMPERATURE
C                           2ND DIMENSION - DENSITY
C
C     ROUTINES:
C     ---------
C         XXOPEN   -
C         XXTERM   -
C         XXSLEN   -
C         XXIN17   -  FETCH DATA FROM MASTER CONDENSED FILE
C         D4SPLN   -  INTERPOLATE CONDENSED MASTER FILE
C                     UPDATED VERSION OF D1SPLN
```

```
C
C-----------------------------------------
C AUTHOR:  LORNE D. HORTON
C          ROOM K1/1/58, JET JOINT UNDERTAKING
C
C   DATE:  23RD FEBRUARY 1996
C-----------------------------------------------------------------------
C UNIX-IDL PORT:
C
C VERSION: 1.1 DATE: 11-11-96
C MODIFIED: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C - FIRST CONVERTED
C VERSION: 1.2 DATE: 20-10-97
C MODIFIED: LORNE HORTON (JET)
C - INCREASED SPACE FOR FILE NAME DIAGNOSTICS
C        - REMOVED ISWIT VARIABLES AND PASSED ITYPE TO
C          D4SPLN INSTEAD
C        - ALLOWED LSWIT TO BE TRUE FOR ALL SCD'S, INCLUDING
C            FROM METASTABLE STATES
C
C VERSION: 1.3 DATE: 13-10-99
C MODIFIED: Martin O'Mullane
C - PRB files wrote incorrect information in IPRT
C                and JPRT line in adf11 file.
C-----------------------------------------------------------------------
C
C
      CHARACTER*8         DATE
      CHARACTER*80        DSNIN(50,10)
      INTEGER             IST2,        IST5,        ITYPE,       IWRITE
      INTEGER             IZ0,         IZH,         IZL,         MAXD
      INTEGER             MAXT,        NGRD(50),    NUDMAX,      NUMMAX
      INTEGER             NUTMAX,      NUZMAX
      REAL*8              DENSA(NUDMAX)
      REAL*8              METFRC(NUDMAX,NUTMAX,NUZMAX,NUMMAX)
      REAL*8              TEK(NUTMAX)
```

5.2 bnd404b: Subroutine bnd404b from library adas4xx

```
       SUBROUTINE BND404B(ITYPE  ,
     &                    NUTMAX , NUDMAX , NUZMAX , NUMMAX ,
     &                    MAXT   , MAXD   ,
     &                    IZL    , IZH    , IZ0    ,
     &                    TEK    , DENSA  ,
     &                    METFRC ,
     &                    NGRD   ,
     &                    IST2   , IST5   , IWRITE , DATE,
     &                    DSNIN)
C
C-----------------------------------------------------------------------
C
C ************ FORTRAN 77 SUBROUTINE BND404B ***************************
C
C   VERSION 1.0
C
C   PURPOSE:
C           TO FETCH DATA FROM RESOLVED ADF10 FILES, SPLINE THEM
C           ONTO THE REQUESTED TEMPERATURE/DENSITY GRID, BUNDLE
C           THEM INTO UNRESOLVED DATA USING THE INPUT METASTABLE
C           FRACTIONS, AND WRITE THE RESULT TO ADF11 FILES.
C
C   CALLING ROUTINE / PROGRAM : LH404RU / ADAS404
C
C   DATA:
C
C           THE SOURCE DATA IS CONTAINED AS MEMBERS OF PARTITIONED
C           DATA SETS AS FOLLOWS:
C
C             8. JETUID.PLT<YR>.DATA
C             9. JETUID.PLS<YR>.DATA
C
C           WHERE <YR> DENOTES TWO INTEGERS FOR THE YEAR SELECTED.
C
C   SUBROUTINE:
C
C   INPUT : (I*4)  ITYPE  - TYPE OF ADF10 DATA BEING READ (SEE ABOVE)
C   INPUT : (I*4)  NUTMAX - OUTPUT ELEMENT MASTER FILE
C                           MAXIMUM NUMBER OF TEMPERATURES
C   INPUT : (I*4)  NUDMAX - OUTPUT ELEMENT MASTER FILE
C                           MAXIMUM NUMBER OF DENSITIES
C   INPUT : (I*4)  NUZMAX - OUTPUT ELEMENT MASTER FILE
C                           MAXIMUM NUMBER OF CHARGE STATES
C   INPUT : (I*4)  NUMMAX - OUTPUT ELEMENT MASTER FILE
C                           MAXIMUM NUMBER OF METASTABLES
C   INPUT : (I*4)  MAXT   - OUTPUT ELEMENT MASTER FILE
C                           ACTUAL NUMBER OF TEMPERATURES
C   INPUT : (I*4)  MAXD   - OUTPUT ELEMENT MASTER FILE
C                           ACTUAL NUMBER OF DENSITIES
C   INPUT : (I*4)  IZL    - LOWEST ION CHARGE TO READ
C   INPUT : (I*4)  IZH    - HIGHEST ION CHARGE TO READ
C   INPUT : (I*4)  IZ0    - NUCLEAR CHARGE TO READ
C   INPUT : (R*8)  DENSA()- OUTPUT ELEMENT MASTER FILE
C                           SET OF MAXD DENSITIES
C   INPUT : (R*8)  TEK()  - OUTPUT ELEMENT MASTER FILE
C                           SET OF MAXT TEMPERATURES
C   INPUT : (R*8)  METFRC(,,,) - METASTABLE POPULATION FRACTIONS,
C                           SPLINED ONTO THE OUTPUT TEMPERATURES
C                           AND DENSITIES
```

```
C                                1ST DIMENSION - DENSITY INDEX
C                                2ND DIMENSION - TEMPERATURE INDEX
C                                3RD DIMENSION - CHARGE STATE INDEX
C                                4TH DIMENSION - METASTABLE INDEX
C   INPUT : (I*4)  NGRD() - NUMBER OF GROUND STATES OF THE FIRST
C                           50 ISOELECTRONIC SEQUENCES
C   INPUT : (I*4)  IST2   - UNIT NUMBER FOR OUTPUT INFORMATION
C                           AND ERROR MESSAGES
C   INPUT : (I*4)  IST5   - UNIT NUMBER FOR READING MASTER CONDENSED
C                           FILE
C   INPUT : (I*4)  IWRITE - UNIT NUMBER FOR WRITING ADF11 DATA
C   INPUT : (C*8)  DATE   - CURRENT DATE
C
C   PARAMETER : (I*4)  NTDMAX - SIZE OF LOCAL WORKING SPACE
C                               (MUST BE GREATER THAN NUTMAX & NUDMAX)
C   PARAMETER : (I*4)  NDZ1V  - MASTER CONDENSED FILE
C                               MAXIMUM NUMBER OF CHARGE STATES
C   PARAMETER : (I*4)  NDTIN  - MASTER CONDENSED FILE
C                               MAXIMUM NUMBER OF TEMPERATURES
C   PARAMETER : (I*4)  NDDEN  - MASTER CONDENSED FILE
C                               MAXIMUM NUMBER OF DENSITIES
C   PARAMETER : (I*4)  NDMET  - MASTER CONDENSED FILE
C                               MAXIMUM NUMBER OF METASTABLES
C
C         : (R*8)  DENSR()    - INPUT MASTER CONDENSED FILE
C                               SET OF IDE REDUCED DENSITIES
C         : (R*8)  TR()       - INPUT MASTER CONDENSED FILE
C                               SET OF ITE REDUCED TEMPERATURES
C         : (R*8)  ZIPT()     - INPUT MASTER CONDENSED FILE
C                               SET OF IZE RECOMBINING ION CHARGES
C         : (R*8)  AIPTM(,,)  - INPUT MASTER CONDENSED FILE
C                               RATIO OF METASTABLE TO GROUND POP.
C                               1ST DIMENSION - DENSITY INDEX
C                               2ND DIMENSION - TEMPERATURE INDEX
C                               3RD DIMENSION - CHARGE STATE INDEX
C                               4TH DIMENSION - METASTABLE INDEX
C         : (R*8)  EIA()      - INPUT MASTER CONDENSED FILE
C                               SET OF IONISATION POTENTIALS (CM-1)
C
C         : (R*8)  ATTY(,)    - WORK SPACE FOR INTERPOLATION
C                             - STORES LOG10(INTERPOLATED VALUES)
C                               1ST DIMENSION - TEMPERATURE
C                               2ND DIMENSION - DENSITY
C         : (R*8)  ARRAY(,)   - STORES LOG10(INTERPOLATED VALUES)
C                               1ST DIMENSION - TEMPERATURE
C                               2ND DIMENSION - DENSITY
C         : (R*8)  WORK(,)    - SUM OF INTERPOLATED VALUES OVER
C                               METASTABLE STATES
C                               1ST DIMENSION - TEMPERATURE
C                               2ND DIMENSION - DENSITY
C
C     ROUTINES:
C     ---------
C         XXOPEN   -
C         XXTERM   -
C         XXIN80   -  FETCH DATA FROM MASTER CONDENSED FILE
C         D4SPLN   -  INTERPOLATE CONDENSED MASTER FILE
C                     UPDATED VERSION OF D1SPLN
C
C-----------------------------------------------
C AUTHOR:  LORNE D. HORTON
```

```
C          ROOM K1/1/58, JET JOINT UNDERTAKING
C
C   DATE:  21ST FEBRUARY 1996
C-----------------------------------------------------------------------
C UNIX-IDL PORT:
C
C VERSION: 1.1 DATE: 11-11-96
C MODIFIED: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C - FIRST CONVERTED
C
C-----------------------------------------------------------------------
C
C VERSION: 1.2 DATE: 20-10-97
C MODIFIED: LORNE HORTON (JET)
C - INCREASED SPACE FOR DIAGNOSTIC PRINTS OF FILE NAMES
C       - ADDED D4OPEN TO SUPPRESS ERROR MESSAGES ON MISSING
C         FILES AND ALLOWED ZERO-FILLING
C       - REMOVED ISWIT VARIABLES AND PASSED ITYPE TO
C         D4SPLN INSTEAD
C       - TURNED LSWIT OFF FOR ALL CASES
C
C-----------------------------------------------------------------------
C
      CHARACTER*8         DATE
      CHARACTER*80        DSNIN(50,10)
      INTEGER             IST2,        IST5,        ITYPE,       IWRITE
      INTEGER             IZ0,         IZH,         IZL,         MAXD
      INTEGER             MAXT,        NGRD(50),    NUDMAX,      NUMMAX
      INTEGER             NUTMAX,      NUZMAX
      REAL*8              DENSA(NUDMAX)
      REAL*8              METFRC(NUDMAX,NUTMAX,NUZMAX,NUMMAX)
      REAL*8              TEK(NUTMAX)
```

5.3 c1u2lc: Subroutine c1u2lc from library adas4xx

```
      FUNCTION C1U2LC(CHR)
C-----------------------------------------------------------------------
C
C  PURPOSE: RETURNS LOWER CASE CHARACTER
C
C  INPUT  (C*1) CHR - INPUT UPPER CASE CHARACTER.
C
C  OUTPUT (C*1) C1U2LC - OUTPUT CORRESPONDING UPPER CASE CHARACTER OF
C  CHR.
C
C  ALESSANDRO LANZAFAME, UNIVERSITY OF STRATHCLYDE, MAR20-95
C
C VERSION 1.1                                          DATE: 27-10-97
C RICHARD MARTIN
C PUT UNDER SCCS CONTROL.
C
C VERSION 1.2                                          DATE: 27-10-97
C ALLAN WHITEFORD
C UPDATED COMMENTS AS PART OF SUBROUTINE DOCUMENTATION
C PROCEDURE
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      CHARACTER           CHR
```

5.4 cnv404a: Subroutine cnv404a from library adas4xx

```
       SUBROUTINE CNV404A(ITYPE  , ISWIT  ,
     &                    NUTMAX , NUDMAX , NUZMAX , NUMMAX ,
     &                    MAXT   , MAXD   ,
     &                    IZL    , IZH    , IZ0    ,
     &                    TEK    , DENSA  ,
     &                    NGRD   ,
     &                    IST2   , IST5   , IWRITE , DATE,
     &                    DSNIN  , OPEN17)
C
C-----------------------------------------------------------------------
C
C ************ FORTRAN 77 SUBROUTINE BND404A ***************************
C
C   VERSION 1.0
C
C   PURPOSE:
C           TO FETCH DATA FROM RESOLVED ADF10 FILES, SPLINE THEM
C           ONTO THE REQUESTED TEMPERATURE/DENSITY GRID, AND
C           WRITE THE RESULT TO ADF11 FILES.
C
C   CALLING ROUTINE / PROGRAM : LH404RR / ADAS404
C
C   DATA:
C
C           THE SOURCE DATA IS CONTAINED AS MEMBERS OF PARTITIONED
C           DATA SETS AS FOLLOWS:
C
C             1. JETUID.ACD<YR>.DATA
C             2. JETUID.SCD<YR>.DATA
C             3. JETUID.CCD<YR>.DATA
C             4. JETUID.PRB<YR>.DATA
C             5. JETUID.PRC<YR>.DATA
C             6. JETUID.QCD<YR>.DATA
C             7. JETUID.XCD<YR>.DATA
C
C           WHERE <YR> DENOTES TWO INTEGERS FOR THE YEAR SELECTED.
C
C   SUBROUTINE:
C
C   INPUT : (I*4)  ITYPE  - TYPE OF ADF10 DATA BEING READ (SEE ABOVE)
C   INPUT : (I*4)  ISWIT  - SWITCH FOR INTERPOLATION TYPE
C   INPUT : (I*4)  NUTMAX - OUTPUT ELEMENT MASTER FILE
C                                MAXIMUM NUMBER OF TEMPERATURES
C   INPUT : (I*4)  NUDMAX - OUTPUT ELEMENT MASTER FILE
C                                MAXIMUM NUMBER OF DENSITIES
C   INPUT : (I*4)  NUZMAX - OUTPUT ELEMENT MASTER FILE
C                                MAXIMUM NUMBER OF CHARGE STATES
C   INPUT : (I*4)  NUMMAX - OUTPUT ELEMENT MASTER FILE
C                                MAXIMUM NUMBER OF METASTABLES
C   INPUT : (I*4)  MAXT   - OUTPUT ELEMENT MASTER FILE
C                                ACTUAL NUMBER OF TEMPERATURES
C   INPUT : (I*4)  MAXD   - OUTPUT ELEMENT MASTER FILE
C                                ACTUAL NUMBER OF DENSITIES
C   INPUT : (I*4)  IZL    - LOWEST ION CHARGE TO READ
C   INPUT : (I*4)  IZH    - HIGHEST ION CHARGE TO READ
C   INPUT : (I*4)  IZ0    - NUCLEAR CHARGE TO READ
C   INPUT : (R*8)  DENSA()- OUTPUT ELEMENT MASTER FILE
C                                SET OF MAXD DENSITIES
C   INPUT : (R*8)  TEK()  - OUTPUT ELEMENT MASTER FILE
```

```
C                                   SET OF MAXT TEMPERATURES
C   INPUT : (I*4)  NGRD() - NUMBER OF GROUND STATES OF THE FIRST
C                           50 ISOELECTRONIC SEQUENCES
C   INPUT : (I*4)  IST2   - UNIT NUMBER FOR OUTPUT INFORMATION
C                           AND ERROR MESSAGES
C   INPUT : (I*4)  IST5   - UNIT NUMBER FOR READING MASTER CONDENSED
C                           FILE
C   INPUT : (I*4)  IWRITE - UNIT NUMBER FOR WRITING ADF11 DATA
C   INPUT : (C*8)  DATE   - CURRENT DATE
C
C   PARAMETER : (I*4)  NTDMAX - SIZE OF LOCAL WORKING SPACE
C                               (MUST BE GREATER THAN NUTMAX & NUDMAX)
C   PARAMETER : (I*4)  NDZ1V  - MASTER CONDENSED FILE
C                               MAXIMUM NUMBER OF CHARGE STATES
C   PARAMETER : (I*4)  NDTIN  - MASTER CONDENSED FILE
C                               MAXIMUM NUMBER OF TEMPERATURES
C   PARAMETER : (I*4)  NDDEN  - MASTER CONDENSED FILE
C                               MAXIMUM NUMBER OF DENSITIES
C
C         : (R*8)  DENSR()    - INPUT MASTER CONDENSED FILE
C                               SET OF IDE REDUCED DENSITIES
C         : (R*8)  TR()       - INPUT MASTER CONDENSED FILE
C                               SET OF ITE REDUCED TEMPERATURES
C         : (R*8)  ZIPT()     - INPUT MASTER CONDENSED FILE
C                               SET OF IZE RECOMBINING ION CHARGES
C         : (R*8)  AIPT(,,)   - INPUT MASTER CONDENSED FILE
C                               RELEVANT RATE COEFFICIENTS
C                               1ST DIMENSION - DENSITY INDEX
C                               2ND DIMENSION - TEMPERATURE INDEX
C                               3RD DIMENSION - CHARGE STATE INDEX
C         : (R*8)  EIA()      - INPUT MASTER CONDENSED FILE
C                               SET OF IONISATION POTENTIALS (CM-1)
C
C         : (R*8)  ATTY(,)    - WORK SPACE FOR INTERPOLATION
C                             - STORES LOG10(INTERPOLATED VALUES)
C                               1ST DIMENSION - TEMPERATURE
C                               2ND DIMENSION - DENSITY
C         : (R*8)  ARRAY(,)   - STORES LOG10(INTERPOLATED VALUES)
C                               1ST DIMENSION - TEMPERATURE
C                               2ND DIMENSION - DENSITY
C
C     ROUTINES:
C     ---------
C         XXOPEN   -
C         XXTERM   -
C         XXIN17   -  FETCH DATA FROM MASTER CONDENSED FILE
C         D4SPLN   -  INTERPOLATE CONDENSED MASTER FILE
C                     UPDATED VERSION OF D1SPLN
C
C-----------------------------------------------
C AUTHOR:  LORNE D. HORTON
C          ROOM K1/1/58, JET JOINT UNDERTAKING
C
C   DATE:   5TH AUGUST 1996
C-----------------------------------------------------------------------
C UNIX-IDL PORT:
C
C VERSION: 1.1 DATE: 11-11-96
C MODIFIED: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C - FIRST CONVERTED
C VERSION: 1.2 DATE: 20-10-97
```

```
C MODIFIED: LORNE HORTON (JET)
C - INCREASED SPACE FOR FILE NAME DIAGNOSTICS
C
C VERSION: 1.3 DATE: 13-10-99
C MODIFIED: MARTIN O'MULLANE
C - PRB DEFINITION HAS BEEN CHANGED AND THEY ARE NOW
C                 SUMMED OVER THE PARENTS. THIS NECESSITATES WRITING
C                 DIFFERENT INFORMATION ON THE HEADER LINE OF THE
C                 ADF11 FILE.
C
C VERSION: 1.4 DATE: 13-10-99
C MODIFIED: MARTIN O'MULLANE
C - DO NOT WRITE QCD AND XCD BLOCKS WHICH ARE ALL ZERO.
C
C VERSION :  1.5
C DATE    : 04-01-2007
C MODIFIED: Martin O'Mullane
C           - When converting from ergs to Joules check for zero value
C             before subtracting (it's a log!) 7. In this case set
C             value of element to -74.0.
C
C-----------------------------------------------------------------------
C
      CHARACTER*8         DATE
      CHARACTER*80        DSNIN(50,10)
      INTEGER             IST2,         IST5,         ISWIT,        ITYPE
      INTEGER             IWRITE,       IZ0,          IZH,          IZL
      INTEGER             MAXD,         MAXT,         NGRD(50),     NUDMAX
      INTEGER             NUMMAX,       NUTMAX,       NUZMAX
      LOGICAL             OPEN17
      REAL*8              DENSA(NUDMAX),              TEK(NUTMAX)
```

5.5 cnv404b: Subroutine cnv404b from library adas4xx

```
       SUBROUTINE CNV404B(ITYPE  , ISWIT  ,
     &                    NUTMAX , NUDMAX , NUZMAX , NUMMAX ,
     &                    MAXT   , MAXD   ,
     &                    IZL    , IZH    , IZ0    ,
     &                    TEK    , DENSA  ,
     &                    NGRD   ,
     &                    IST2   , IST5   , IWRITE , DATE,
     &                    DSNIN  , OPEN17)
C
C-----------------------------------------------------------------------
C
C ************ FORTRAN 77 SUBROUTINE BND404B ***************************
C
C   VERSION 1.0
C
C   PURPOSE:
C           TO FETCH DATA FROM RESOLVED ADF10 FILES, SPLINE THEM
C           ONTO THE REQUESTED TEMPERATURE/DENSITY GRID, AND
C           WRITE THE RESULT TO ADF11 FILES.
C
C   CALLING PROGRAM LH404RR
C
C   DATA:
C
C           THE SOURCE DATA IS CONTAINED AS MEMBERS OF PARTITIONED
C           DATA SETS AS FOLLOWS:
C
C             8. JETUID.PLT<YR>.DATA
C             9. JETUID.PLS<YR>.DATA
C
C           WHERE <YR> DENOTES TWO INTEGERS FOR THE YEAR SELECTED.
C
C   SUBROUTINE:
C
C   INPUT : (I*4)  ITYPE  - TYPE OF ADF10 DATA BEING READ (SEE ABOVE)
C   INPUT : (I*4)  ISWIT  - SWITCH FOR INTERPOLATION TYPE
C   INPUT : (I*4)  NUTMAX - OUTPUT ELEMENT MASTER FILE
C                                 MAXIMUM NUMBER OF TEMPERATURES
C   INPUT : (I*4)  NUDMAX - OUTPUT ELEMENT MASTER FILE
C                                 MAXIMUM NUMBER OF DENSITIES
C   INPUT : (I*4)  NUZMAX - OUTPUT ELEMENT MASTER FILE
C                                 MAXIMUM NUMBER OF CHARGE STATES
C   INPUT : (I*4)  NUMMAX - OUTPUT ELEMENT MASTER FILE
C                                 MAXIMUM NUMBER OF METASTABLES
C   INPUT : (I*4)  MAXT   - OUTPUT ELEMENT MASTER FILE
C                                 ACTUAL NUMBER OF TEMPERATURES
C   INPUT : (I*4)  MAXD   - OUTPUT ELEMENT MASTER FILE
C                                 ACTUAL NUMBER OF DENSITIES
C   INPUT : (I*4)  IZL    - LOWEST ION CHARGE TO READ
C   INPUT : (I*4)  IZH    - HIGHEST ION CHARGE TO READ
C   INPUT : (I*4)  IZ0    - NUCLEAR CHARGE TO READ
C   INPUT : (R*8)  DENSA()- OUTPUT ELEMENT MASTER FILE
C                                 SET OF MAXD DENSITIES
C   INPUT : (R*8)  TEK()  - OUTPUT ELEMENT MASTER FILE
C                                 SET OF MAXT TEMPERATURES
C   INPUT : (I*4)  NGRD() - NUMBER OF GROUND STATES OF THE FIRST
C                           50 ISOELECTRONIC SEQUENCES
C   INPUT : (I*4)  IST2   - UNIT NUMBER FOR OUTPUT INFORMATION
C                           AND ERROR MESSAGES
```

```
C   INPUT : (I*4)  IST5   - UNIT NUMBER FOR READING MASTER CONDENSED
C                           FILE
C   INPUT : (I*4)  IWRITE - UNIT NUMBER FOR WRITING ADF11 DATA
C   INPUT : (C*8)  DATE   - CURRENT DATE
C
C   PARAMETER : (I*4)  NTDMAX - SIZE OF LOCAL WORKING SPACE
C                               (MUST BE GREATER THAN NUTMAX & NUDMAX)
C   PARAMETER : (I*4)  NDZ1V  - MASTER CONDENSED FILE
C                               MAXIMUM NUMBER OF CHARGE STATES
C   PARAMETER : (I*4)  NDTIN  - MASTER CONDENSED FILE
C                               MAXIMUM NUMBER OF TEMPERATURES
C   PARAMETER : (I*4)  NDDEN  - MASTER CONDENSED FILE
C                               MAXIMUM NUMBER OF DENSITIES
C   PARAMETER : (I*4)  NDMET  - MASTER CONDENSED FILE
C                               MAXIMUM NUMBER OF METASTABLES
C
C         : (R*8)  DENSR()    - INPUT MASTER CONDENSED FILE
C                               SET OF IDE REDUCED DENSITIES
C         : (R*8)  TR()       - INPUT MASTER CONDENSED FILE
C                               SET OF ITE REDUCED TEMPERATURES
C         : (R*8)  ZIPT()     - INPUT MASTER CONDENSED FILE
C                               SET OF IZE RECOMBINING ION CHARGES
C         : (R*8)  AIPTM(,,)  - INPUT MASTER CONDENSED FILE
C                               RATIO OF METASTABLE TO GROUND POP.
C                               1ST DIMENSION - DENSITY INDEX
C                               2ND DIMENSION - TEMPERATURE INDEX
C                               3RD DIMENSION - CHARGE STATE INDEX
C                               4TH DIMENSION - METASTABLE INDEX
C         : (R*8)  EIA()      - INPUT MASTER CONDENSED FILE
C                               SET OF IONISATION POTENTIALS (CM-1)
C
C         : (R*8)  ATTY(,)    - WORK SPACE FOR INTERPOLATION
C                             - STORES LOG10(INTERPOLATED VALUES)
C                               1ST DIMENSION - TEMPERATURE
C                               2ND DIMENSION - DENSITY
C         : (R*8)  ARRAY(,)   - STORES LOG10(INTERPOLATED VALUES)
C                               1ST DIMENSION - TEMPERATURE
C                               2ND DIMENSION - DENSITY
C
C     ROUTINES:
C     ---------
C         XXOPEN   -
C         XXTERM   -
C         XXIN80   -  FETCH DATA FROM MASTER CONDENSED FILE
C         D4SPLN   -  INTERPOLATE CONDENSED MASTER FILE
C                     UPDATED VERSION OF D1SPLN
C
C-----------------------------------------
C AUTHOR:  LORNE D. HORTON
C          ROOM K1/1/58, JET JOINT UNDERTAKING
C
C   DATE:   5TH AUGUST 1996
C-----------------------------------------------------------------------
C UNIX-IDL PORT:
C
C VERSION: 1.1 DATE: 11-11-96
C MODIFIED: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C - FIRST CONVERTED
C
C-----------------------------------------------------------------------
C
```

```
C VERSION: 1.2 DATE: 20-10-97
C MODIFIED: LORNE HORTON (JET)
C - INCREASED SPACE FOR DIAGNOSTIC PRINTS OF FILE NAMES
C       - ADDED D4OPEN TO SUPPRESS ERROR MESSAGES ON MISSING
C         FILES AND ALLOWED ZERO-FILLING
C
C VERSION: 1.3 DATE: 13-10-99
C MODIFIED: Martin O'Mullane
C - PRB definition has been changed and they are now
C                 summed over the parents. This necessitates writing
C                 different information on the header line of the
C                 adf11 file. Format statement 1207 and write
C                 associated with it have been changed.
C
C VERSION :  1.4
C DATE    : 04-01-2007
C MODIFIED: Martin O'Mullane
C           - When converting from ergs to Joules check for zero value
C             before subtracting (it's a log!) 7. In this case set
C             value of element to -74.0.
C
C-----------------------------------------------------------------------
C
      CHARACTER*8         DATE
      CHARACTER*80        DSNIN(50,10)
      INTEGER             IST2,         IST5,         ISWIT,        ITYPE
      INTEGER             IWRITE,       IZ0,          IZH,          IZL
      INTEGER             MAXD,         MAXT,         NGRD(50),     NUDMAX
      INTEGER             NUMMAX,       NUTMAX,       NUZMAX
      LOGICAL             OPEN17
      REAL*8              DENSA(NUDMAX),              TEK(NUTMAX)
```

5.6 cstrl: Subroutine cstrl from library adas4xx

```
      FUNCTION CSTRL(L)
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 FUNCTION: CSTRL  ********************
C
C  PURPOSE: RETURNS CHARACTER CODE FOR L VALUE
C
C
C  INPUT :  INTEGER  L       = Angular Momentumn (eg.  1 )
C
C  OUTPUT:  (C*1)    CSTRL   = Angular Momentumn (eg. 'P')
C
C  Returns a blank string if L > 15
C
C  SUBROUTINES:  NONE
C
C  AUTHOR:  Martin O'Mullane, JET
C
C  DATE:    2/10/97
C
C VERSION:
C MODIFIED:
C
C-----------------------------------------------------------------------
      INTEGER             L
```

5.7 d1spln: Subroutine d1spln from library adas4xx

```
      SUBROUTINE D1SPLN( ISWIT  , LSWIT  ,
     &                   NTDMAX ,
     &                   NDDEN  , NDTIN  , NDZ1V  ,
     &                   ITDVAL ,
     &                   IDE    , ITE    , IZE    ,
     &                   DUSR   , TUSR   , IZ1    ,
     &                   DENSR  , TR     , ZIPT   ,
     &                   EIA    , AIPT   ,
     &                   LZRNG  , LDRNG  , LTRNG  ,
     &                   AOUT
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: D1SPLN *********************
C
C  PURPOSE: TO INTERPOLATE/EXTRAPOLATED DATA FROM MASTER CONDENSED  FILE
C           TO THE USER ENTERED TEMPERATURE/DENSITY PAIRS FOR THE SELEC-
C           TED RECOMBINING ION CHARGE.
C
C  CALLING PROGRAM: ADAS401
C
C  DATA:
C
C           THE  SOURCE  DATA IS CONTAINED AS  MEMBERS  OF  PARTITIONED
C           DATA SETS AS FOLLOWS:
C
C             1. JETUID.ACD<YR>.DATA
C             2. JETUID.SCD<YR>.DATA
C             3. JETUID.CCD<YR>.DATA
C             4. JETUID.PRB<YR>.DATA
C             5. JETUID.PRC<YR>.DATA
C             6. JETUID.PRB<YR>.DATA
C             7. JETUID.PRC<YR>.DATA
C             8. JETUID.PLT<YR>.DATA
C             9. JETUID.PLS<YR>.DATA
C            10. JETUID.MET<YR>.DATA
C
C           WHERE <YR> DENOTES TWO INTEGERS FOR THE YEAR SELECTED.
C
C           THE PARTICULAR TYPE OPENED (1-10) IS SELECTED BY 'ISWIT'
C
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  ISWIT   = DATA TYPE SELECTOR (SEE ABOVE) (1 -> 10)
C  INPUT : (L*4)  LSWIT   = .TRUE.  => SET OF 'EIA' VALUES PRESENT IN
C                                      MASTER CONDENSED FILE.
C                           .FALSE  => SET OF 'EIA' VALUES NOT PRESENT
C                                      IN MASTER CONDENSED FILE.
C
C  INPUT : (I*4)  NTDMAX  = USER ENTERED VALUES -
C                           MAXIMUM NUMBER OF TEMPERATURE/DENSITY PAIRS
C
C  INPUT : (I*4)  NDDEN   = INPUT MASTER CONDENSED FILE -
C                           MAXIMUM NUMBER OF REDUCED DENSITIES
C  INPUT : (I*4)  NDTIN   = INPUT MASTER CONDENSED FILE -
C                           MAXIMUM NUMBER OF REDUCED TEMPERATURES
C  INPUT : (I*4)  NDZ1V   = INPUT MASTER CONDENSED FILE -
C                           MAXIMUM NUMBER OF CHARGE STATES
```

```
C
C  INPUT : (I*4)  ITDVAL  = USER ENTERED VALUES -
C                            NUMBER OF TEMPERATURE/DENSITY PAIRS ENTERED
C
C  INPUT : (I*4)  IDE     = INPUT MASTER CONDENSED FILE -
C                            NUMBER OF REDUCED DENSITIES READ
C  INPUT : (I*4)  ITE     = INPUT MASTER CONDENSED FILE -
C                            NUMBER OF REDUCED TEMPERATURES READ
C  INPUT : (I*4)  IZE     = INPUT MASTER CONDENSED FILE -
C                            NUMBER OF CHARGE STATES READ
C
C  INPUT : (R*8)  DUSR()  = USER ENTERED VALUES -
C                           SET OF 'ITDVAL' ENTERED ELECTRON DENSITIES
C                           (UNITS: CM**-3)
C                           DIMENSION: TEMPERATURE/DENSITY PAIR INDEX
C  INPUT : (R*8)  TUSR()  = USER ENTERED VALUES -
C                           SET OF'ITDVAL' ENTERED ELECTRON TEMPERATURES
C                           (UNITS: KELVIN)
C                           DIMENSION: TEMPERATURE/DENSITY PAIR INDEX
C  INPUT : (I*4)  IZ1     = USER ENTERED VALUE -
C                           RECOMBINING ION CHARGE
C
C  INPUT : (R*8)  DENSR() = INPUT MASTER CONDENSED FILE -
C                           SET OF 'IDE' REDUCED DENSITIES (CM-3/Z1**7)
C  INPUT : (R*8)  TR()    = INPUT MASTER CONDENSED FILE -
C                           SET OF 'ITE' REDUCED TEMPERATURES (K/Z1**2)
C  INPUT : (R*8)  ZIPT()  = INPUT MASTER CONDENSED FILE -
C                           SET OF 'IZE' INPUT RECOMBINING ION CHARGES
C
C  INPUT : (R*8)  EIA()   = INPUT MASTER CONDENSED FILE -
C                           IONISATION RATE COEFFTS. - (UNITS: RYDBERGS)
C                           DIMENSION: ION CHARGE
C  INPUT : (R*8)  AIPT(,,)= INPUT MASTER CONDENSED FILE -
C                           RELEVANT COEFFICIENT/POWER DATA FOR 'ISWIT'.
C                           1ST DIMENSION: DENSITY INDEX     ('DENSR()')
C                           2ND DIMENSION: TEMPERATURE INDEX  ('TR()')
C                           3RD DIMENSION: CHARGE STATE INDEX ('ZIPT()')
C
C  OUTPUT: (L*4)  LZRNG(1) = .TRUE.  => 'AOUT()' VALUES FOR CHARGE-
C                                       STATE 'IZ1' INTERPOLATED.
C                          = .FALSE. => 'AOUT()' VALUE  FOR CHARGE-
C                                       STATE 'IZ1' EXTRAPOLATED.
C  OUTPUT: (L*4)  LDRNG()  = .TRUE.  => 'AOUT()' VALUE FOR DENSITY
C                                       INDEX INTERPOLATED.
C                          = .FALSE. => 'AOUT()' VALUE FOR DENSITY
C                                       INDEX EXTRAPOLATED.
C                           DIMENSION: DENSITY INDEX
C  OUTPUT: (L*4)  LTRNG()  = .TRUE.  => 'AOUT()' VALUE FOR TEMPERATURE
C                                       INDEX INTERPOLATED.
C                          = .FALSE. => 'AOUT()' VALUE FOR TEMPERATURE
C                                       INDEX EXTRAPOLATED.
C                           DIMENSION: TEMPERATURE INDEX
C
C  OUTPUT: (R*8)  AOUT()   = EXTRAPOLATED/INTERPOLATED  DATA  FOR  EACH
C                            USER ENTERED TEMPERATURE/DENSITY PAIR.
C                            DIMENSION: TEMPERATURE/DENSITY PAIR INDEX.
C
C          (I*4)  NUDIM   = PARAMETER = MUST BE GREATER THAN OR EQUAL TO
C                                       'NTDMAX'.
C
C          (I*4)  NDMAX1  = 'NDDEN'
```

```
C          (I*4)  NTMAX1  = 'NDTIN'
C          (I*4)  NZMAX1  = 'NDZ1V'
C          (I*4)  NDMAX2  = 'NUDIM'
C          (I*4)  NTMAX2  = 'NUDIM'
C          (I*4)  ITD     = GENERAL USE ARRAY SUBSCRIPT INDEX
C
C          (R*8)  ATTY(,) = WORKING SPACE FOR 3-WAY SPLINE ITERPOLATION
C                           ( STORES LOG10(INTERPOLATED VALUES) )
C                            1ST DIMENSION: TEMPERATURE
C                            2ND DIMENSION: DENSITY
C
C NOTE:
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          DXSPL1     ADAS      1ST PART OF 3-WAY SPLINE OF INPUT DATA
C          DXSPL2     ADAS      2ND PART OF 3-WAY SPLINE OF INPUT DATA
C          DXSPL3     ADAS      3RD PART OF 3-WAY SPLINE OF INPUT DATA
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/37
C          JET EXT. 2516
C
C DATE:    17/06/91
C
C UNIX PORT:
C
C VERSION: 1.1                          DATE: 06-09-95
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C             -  FIRST RELEASE
C
C VERSION: 1.2                          DATE: 15-04-96
C MODIFIED: TIM HAMMOND/PAUL BRIDEN
C             -  CORRECT BUG: SET NDMAX2 AND NTMAX2 TO NUDIM AND NOT
C                NTDMAX.
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IDE,          ISWIT,       ITDVAL,      ITE
      INTEGER             IZ1,          IZE,         NDDEN,       NDTIN
      INTEGER             NDZ1V,        NTDMAX
      LOGICAL             LDRNG(NTDMAX),             LSWIT
      LOGICAL             LTRNG(NTDMAX),             LZRNG(1)
      REAL*8              AIPT(NDDEN,NDTIN,NDZ1V),   AOUT(NTDMAX)
      REAL*8              DENSR(NDDEN),              DUSR(NTDMAX)
      REAL*8              EIA(50),      TR(NDTIN),   TUSR(NTDMAX)
      REAL*8              ZIPT(NDZ1V)
```

5.8 d1titl: Subroutine d1titl from library adas4xx

```
      SUBROUTINE D1TITL( IMSEL  , DSFULL ,
     &                   IPRNT  , IPSYS  ,
     &                   CELEM  , CYEAR  , CLASS  ,
     &                   TITLX
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: D1TITL *********************
C
C  PURPOSE:  TO CREATE THE DESCRIPTIVE TITLE FOR SELECTED DATA.
C
C  CALLING PROGRAM: ADAS401
C
C  SUBROUTINE:
C
C
C  INPUT : (I*4)  IMSEL    = INDEX OF SELECTED METASTABLE STATE
C                            (ONLY USED IF (LPARTL.AND.LPOWER) -
C                             EQUALS ONE OTHERWISE).
C  INPUT : (C*80) DSFULL   = FULL MVS INPUT DATA SET NAME
C
C  INPUT : (I*4)  IPRNT    = PARENT INDEX FOR SELECTED 'METASTABLE' INDEX
C                            NOT USED FOR STANDARD MASTER CONDENSED FILES
C  INPUT : (I*4)  IPSYS    = SPIN SYSTEM  FOR SELECTED 'METASTABLE' INDEX
C                            NOT USED FOR STANDARD MASTER CONDENSED FILES
C
C  INPUT : (C*2)  CELEM    = ISO-ELECTRONIC SEQUENCE ELEMENT NAME
C  INPUT : (C*2)  CYEAR    = YEAR OF INPUT DATA (2 CHARACTER ABBREV.)
C
C  INPUT : (C*3)  CLASS    = CLASS OF DATA SET BEING ANALYSED.
C                            ('ACD','SCD','CCD','PRB','PRC','PLT','PLS')
C
C  OUTPUT: (C*80) TITLX     = SELECTED DATA-BLOCK: DESCRIPTIVE TITLE
C
C          (C*1)  CPRNT     = 'IPRNT' AS CHARACTER
C          (C*1)  CPSYS     = 'IPSYS' AS CHARACTER
C          (C*2)  CMSEL     = 'IMSEL' AS CHARACTER
C
C ROUTINES: NONE
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/37
C          JET EXT. 2520
C
C DATE:    17/06/91
C
C UNIX PORT:
C
C VERSION: 1.1                          DATE: 06-09-95
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST RELEASE
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      CHARACTER*2         CELEM
      CHARACTER*3         CLASS
      CHARACTER*2         CYEAR
      CHARACTER*80        DSFULL,      TITLX
      INTEGER             IMSEL,       IPRNT,       IPSYS
```

5.9 d2data: Subroutine d2data from library adas4xx

```
       SUBROUTINE D2DATA( DSFULL , TITLF  , IFAIL  , LRESO ,IPRT , IGRND
     &                  , IZ0    , IZ1    , ICLASS , ITMAX , IEVCUT
     &                  , ITDIMD , ITMAXD , IDMAXD , IZMAXD
     &                  , DTEV   , DDENS
     &                  , DTEVD  , DDENSD , DRCOFD , ZDATA
     &                  , DRCOFI , YEAR
     &                  )

C-----------------------------------------------------------------------
C
C PURPOSE : TO EXTRACT adf11 COLLISIONAL DIELECTRONIC DATA
C
C NOTE    : THE SOURCE DATA IS CONTAINED IN CENTRAL (OR USER) ADAS
C           DATASETS AS:
C           /home/adas/adas/adf11/<typ><yr>/<typ><yr>_<el>.<fl>.dat
C
C           where, <yr>  = nominal year of data
C                  <el>  = element name
C                  <typ> = type of data
C                  <fl>  = optional filter for power data
C
C           The classes are
C              iclass = 1 : acd
C              iclass = 2 : scd
C              iclass = 3 : ccd
C              iclass = 4 : prb
C              iclass = 5 : prc
C              iclass = 6 : plt
C              iclass = 7 : pls
C
C
C           This routine was originally used to extract data for the SANCO
C           impurity transport code where filenames were constructed from
C           a userid, the year and a cutoff energy. The routine was
C           rewritten so that the full adf11 filename was passed. Hence some
C           of the inputs are unecessary but are retained for backwards
C           compatibility in calling codes.
C
C INPUT  : (C*2)  YEAR      = YEAR OF DATA (not used)
C          (C*2)  YEARDF    = DEFAULT YEAR OF DATA IF REQUESTED YEAR
C                             DOES NOT EXIST
C          (I*4)  IZ0       = NUCLEAR CHARGE
C          (I*4)  IZ1       = MINIMUM ION CHARGE + 1 (not used)
C          (I*4)  ICLASS    = CLASS OF DATA (1 - 6)
C          (I*4)  ITMAX     = NUMBER OF ( DTEV() , DDENS() ) PAIRS
C          (I*4)  IEVCUT    = ENERGY CUT-OFF (EV) (not used)
C          (R*8)  DTEV()    = DLOG10(ELECTRON TEMPERATURES (EV))
C          (R*8)  DDENS()   = DLOG10(ELECTRON DENSITIES (CM-3))
C          (I*4)  IPRT      = INDEX OF PARENT STATE
C          (I*4)  IGRND     = INDEX OF GROUND STATE
C          (L*4)  LRESO     = FLAG WHETHER RESOLVED DATA
C
C OUTPUT : (C*120)TITLF     = INFORMATION STRING
C          (I*4)  ITDIMD    = MAXIMUM NUMBER OF DATA TEMP & DENS
C          (I*4)  ITMAXD    = NUMBER OF DATA DTEVD()
C          (I*4)  IDMAXD    = NUMBER OF DATA DDENS()
C          (I*4)  IZMAXD    = NUMBER OF DATA ZDATA()
C          (I*4)  ITDIMD    = MAXIMUM NUMBER OF DATA TEMP & DENS
C          (I*4)  ZDATA()   = Z1 CHARGES IN DATASET
```

```
C          (I*4)  IFAIL     = -1   IF ROUTINE SUCCESSFUL BUT THE DEFAULT
C                                   YEAR FOR THE DATA WAS USED.
C                           = 0    IF ROUTINE SUCCESSFUL - DATA FOT THE
C                                   REQUESTED YEAR USED.
C                           = 1    IF ROUTINE OPEN STATEMENT FAILED
C          (R*8)  DTEVD()   = DLOG10(DATA ELECTRON TEMPERATURES (EV))
C          (R*8)  DDENSD()  = DLOG10(DATA ELECTRON DENSITIES (CM-3))
C          (R*8)  DRCOFD()  = DLOG10(DATA RATE COEFFICIENTS (CM-3/S))
C          (R*8)  DRCOFI()  = INTERPOLATION OF DRCOFD(,,) FOR
C                             DTEV() & DDENS()
C
C PROGRAM: (C*80) STRING    = GENERAL VARIABLE
C          (C*80) BLANK     = BLANK STRING
C          (C*2)  YEARSV    = LAST YEAR USED IN THIS ROUTINE
C          (I*4)  IREAD     = INPUT STREAM FOR OPEN STATEMENT
C          (I*4)  IZ0SV     = LAST IZ0 USED IN THIS ROUTINE
C          (I*4)  ICLSV     = LAST ICLASS USED IN THIS ROUTINE
C          (I*4)  INDXZ1    = LOCATION OF IZ1 IN ZDATA()
C          (I*4)  LCK       = MUST BE GREATER THAN 'ITMAXD' & 'IDMAXD'
C                             & 'ITMAX' - ARRAY SIZE FOR SPLINE CALCS.
C          (R*8)  A()       = GENERAL ARRAY
C          (R*8)  DRCOF0(,) = INTERPOLATION OF DRCOFD(,,) W.R.T DTEV()
C          (L*8)  LEXIST    = TRUE --- FILE TO OPEN EXISTS ELSE NOT
C
C PE BRIDEN = ADDED VARIABLES (14/01/91)
C
C          (I*4)  L1      = PARAMETER = 1
C          (I*4)  IOPT    = DEFINES THE BOUNDARY DERIVATIVES FOR THE
C                             SPLINE ROUTINE 'XXSPLE', SEE 'XXSPLE'.
C
C          (L*4)  LSETX   = .TRUE.  => SET UP SPLINE PARAMETERS RELATING
C                                      TO X-AXIS.
C                           .FALSE. => DO NOT SET UP SPLINE PARAMETERS
C                                      RELATING TO X-AXIS.
C                                      (I.E. THEY WERE SET IN A PREVIOUS
C                                            CALL )
C                           (VALUE SET TO .FALSE. BY 'XXSPLE')
C
C
C          (R*8)  DY()    = SPLINE INTERPOLATED DERIVATIVES
C
C          (R*8 ADAS FUNCTION - 'R8FUN1' ( X -> X) )
C
C PE BRIDEN = ADDED VARIABLES (23/04/93)
C
C          (I*4 ADAS FUNCTION - 'I4UNIT' (OUTPUT STREAM))
C
C AUTHOR : JAMES SPENCE (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/80
C          JET  EXT. 4866
C
C DATE   : 22/02/90
C
C DATE   : 21/08/90 PE BRIDEN - REVISION: SEQUA(43) CHANGED ('TE'->'TC')
C
C DATE   : 08/10/90 PE BRIDEN - REVISION: RENAMED SUBROUTINE
C
C DATE   : 12/11/90 PE BRIDEN - CORRECTION: MOVE THE SETTING OF 'INDXZ1'
C                                           TO AFTER THE  '20 CONTINUE'
C                                           STATEMENT.   ALSO SAVE  THE
C                                           VALUE OF 'IZ1MIN'.
```

```
C
C DATE   : 14/01/91 PE BRIDEN - ADAS91:     CALLS TO NAG SPLINE ROUTINES
C                                           'E01BAF' & 'E02BBF' REPLACED
C                                           BY  CALLS   TO  ADAS  SPLINE
C                                           ROUTINE 'XXSPLE'.
C
C DATE   : 25/06/91 PE BRIDEN - CORRECTION: CHANGED FOLLOWING DIMENSION:
C                                            'DIMENSION DRCOFI(ITDIMD)'
C                                           TO
C                                            'DIMENSION DRCOFI(ITMAX)'
C
C DATE   : 07/08/91 PE BRIDEN - ADDED ERROR HANDLING IF THE OPEN STATE-
C                               MENT FAILS. (IFAIL=1 RETURNED)
C
C DATE   : 27/04/92 PE BRIDEN - ADDED DEFAULT YEAR FOR DATA IF REQUESTED
C                               YEAR DOES NOT EXIST. (ADDED 'YEARDF')
C                               INTRODUCED IFAIL = -1 IF DEFAULT YEAR
C                               WAS USED AND NOT THE REQUESTED YEAR.
C
C DATE   : 10/03/93 PE BRIDEN - ALLOWED INPUT DATA SETS TO BE ACCESSED
C                               FROM ANY USERID (DEFAULT = JETSHP)
C                               - INTRODUCED USERID VARIABLE AND CALL
C                                 TO XXUID.
C
C DATE   : 23/04/93 PE BRIDEN - ADDED I4UNIT FUNCTION TO WRITE
C                               STATEMENTS FOR SCREEN MESSAGES
C
C UPDATE:  24/05/93 - PE BRIDEN - ADAS91: CHANGED I4UNIT(0)-> I4UNIT(-1)
C
C UPDATE:  14/09/94 - PE BRIDEN - ADAS91: ADDED CHECK TO MAKE SURE THAT
C                                         ITMAX, ITMAXD AND IDMAXD ARE
C                                         IN RANGE (I.E. <= LCK).
C
C UPDATE:  16/08/96 - PE BRIDEN - ADAS91: MINOR MOD - IF DEFAULT DATA
C                                         IS NOT FOUND THEN ASSIGN TITLF
C                                         BEFORE EXITING WITH AN ERROR.
C
C-----------------------------------------------------------------------
C
C UNIX-IDL PORT:
C
C VERSION: 1.1 DATE: 28-10-96
C MODIFIED: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C - FIRST CONVERTED
C
C VERSION: 1.2 DATE: 14-02-97
C MODIFIED: RICHARD MARTIN
C - CHANGED DEFINITION OF 'BLANKS' TO 80 CHARACTERS ONLY
C
C VERSION: 1.3 DATE: 26-10-97
C MODIFIED: LORNE HORTON (JET)
C - CHANGED EXTRAPOLATION ALGORITHM
C
C VERSION: 1.4 DATE: 23-07-99
C MODIFIED: Martin O'Mullane (JET)
C - Changed the test for checking whether the file was opened.
C               - Removed commented out code for constructing filenames
C               - Modified the comments at the top of the file
C               - Converted code to implicit none
C               - Increased LCK to 200
C-----------------------------------------------------------------------
```

```
C-----------------------------------------------------------------------

C-----------------------------------------------------------------------
      CHARACTER*80        DSFULL
      CHARACTER*120       TITLF
      CHARACTER*2         YEAR
      INTEGER             ICLASS,      IDMAXD,      IEVCUT,      IFAIL
      INTEGER             IGRND,       IPRT,        ITDIMD,      ITMAX
      INTEGER             ITMAXD,      IZ0,         IZ1,         IZMAXD
      LOGICAL             LRESO
      REAL*8              DDENS(ITMAX),             DDENSD(ITDIMD)
      REAL*8              DRCOFD(ITDIMD,ITDIMD,ITDIMD)
      REAL*8              DRCOFI(ITMAX),            DTEV(ITMAX)
      REAL*8              DTEVD(ITDIMD),            ZDATA(ITDIMD)
```

5.10 d2tdin: Subroutine d2tdin from library adas4xx

```
      SUBROUTINE D2TDIN( DSFULL
     &                 , ITDIMD , ITMAXD , IDMAXD
     &                 , TEMP   , DENS   , LRESO
     &                 )
C
C-----------------------------------------------------------------------
C
C PURPOSE : TO EXTRACT TEMPERATURE AND DENSITY FROM ADF11 FILE FOR
C           DISPLAY ON ADAS402 PROCESSING SCREEN
C
C INPUT  : (C*80) DSFULL    = INPUT DATASET NAME
C          (I*4)  ITDIMD    = MAXIMUM NUMBER OF DATA TEMP & DENS
C          (L*4)  LRESO     = .TRUE. => RESOLVED DATASET
C                             .FALSE.=> UNRESOLVED DATASET
C
C OUTPUT : (I*4)  ITMAXD    = NUMBER OF DATA TEMPS
C          (I*4)  IDMAXD    = NUMBER OF DATA DENS
C          (R*8)  TEMP()    = DATA ELECTRON TEMPERATURES (EV)
C          (R*8)  DENS()    = DATA ELECTRON DENSITIES (CM-3)
C
C PROGRAM:
C
C-----------------------------------------------------------------------
C
C UNIX-IDL PORT:
C
C VERSION: 1.1 DATE: 29-10-96
C MODIFIED: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C - FIRST WRITTEN
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
C
      LOGICAL LRESO
C
      CHARACTER*80        DSFULL
      INTEGER             IDMAXD,      ITDIMD,      ITMAXD
      LOGICAL             LRESO
      REAL*8              DENS(ITDIMD),             TEMP(ITDIMD)
```

5.11 d4data: Subroutine d4data from library adas4xx

```
      SUBROUTINE D4DATA( TITLE , DATE  , UIDIN  , USERID ,
     &                   ISWIT , NIND  , YEAR   , SELTAB , REPTAB ,
     &                   NDZ   , NDDEN , NDTIN  ,
     &                   IZ0   , IZ1   , NEL1   , IZ2    , NEL2   ,
     &                   T     , TL    , MAXT   ,
     &                   DENSA , DENSL , MAXD   ,
     &                   ATTY  , ARRAY , ZINTRP , DINTRP , TINTRP ,
     &                   DSNIN , DSNO  , OPEN17
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: D4DATA *********************
C
C  PURPOSE: TO OPEN/ACQUIRE DATA FROM STD.MASTER CONDENSED COLLISIONAL-
C           DIELECTRONIC   FILES,   OBTAIN  INTERPOLATED   COLLISIONAL-
C           DIELECTRONIC RECOMBINATION AND IONISATION COEFFICIENTS, AND
C           PREPARE ELEMENT MASTER FILE IF REQUIRED.
C
C  CALLING PROGRAM: ADAS404
C
C  DATA:
C
C       INPUT:
C       ------
C           THE  SOURCE  DATA IS CONTAINED AS  MEMBERS  OF  PARTITIONED
C           DATA SETS AS FOLLOWS:
C
C             1. JETUID.ACD<YR>.DATA
C             2. JETUID.SCD<YR>.DATA
C             3. JETUID.CCD<YR>.DATA
C             4. JETUID.PRB<YR>.DATA
C             5. JETUID.PRC<YR>.DATA
C             6. JETUID.PLT<YR>.DATA
C             7. JETUID.PLS<YR>.DATA
C
C           WHERE <YR> DENOTES TWO INTEGERS FOR THE YEAR SELECTED.
C           IF <YR> IS BLANK THEN THE CURRENT RECOMMENDED DATA SETS ARE
C           USED
C
C           'JETUID' IS GIVEN BY 'UIDIN'
C
C           THE PARTICULAR TYPE OPENED (1-7) IS SELECTED BY 'ISWIT'
C
C           THE MEMBERS OF THE PARTITIONED DATA SETS ARE  <SE>
C           WHERE <SE> IS THE ONE OR TWO LETTER ION SEQUENCE CODE
C
C           THIS PROGRAM ASSESSES ONLY STANDARD MASTER CONDENSED FILES.
C           ----------------------------------------------------------
C
C
C      OUTPUT:
C      -------
C           THE OUTPUT ELEMENT MASTER DATA IS IN SEQUENTIAL FILES AS
C           FOLLOWS:
C
C             1. JETUID.ACD<YR>#<EL>.DATA
C             2. JETUID.SCD<YR>#<EL>.DATA
C             3. JETUID.CCD<YR>#<EL>.DATA
C             4. JETUID.PRB<YR>#<EL>.DATA
```

```
C             5. JETUID.PRC<YR>#<EL>.DATA
C             6. JETUID.PLT<YR>#<EL>.DATA
C             7. JETUID.PLS<YR>#<EL>.DATA
C
C           WHERE <YR> IS AS ABOVE AND  <EL> IS THE ELEMENT SYMBOL
C
C           'JETUID' IS GIVEN BY 'USERID'
C
C  SUBROUTINE:
C
C  INPUT : (C*32) TITLE    = USER ENTERED PROGRAM RUN TITLE
C  INPUT : (C*8)  DATE     = CURRENT DATE (AS 'DD/MM/YY')
C  INPUT : (C*6)  UIDIN   = PROJECT UID FOR INPUT CONDENSED DATA FILE
C  INPUT : (C*6)  USERID  = USER ID FOR OUTPUT INC. ELEMENT MASTER FILE
C
C  INPUT : (I*4)  ISWIT    = DATA TYPE SELECTOR (SEE ABOVE) (1 -> 7)
C  INPUT : (I*4)  NIND     = NUMBER OF STAGES BEGINNING WITH LOWEST TO
C                            BE SKIPPED.   IT IS ASSUMED THAT DATA FOR
C                            THESE STAGES WILL BE SUPPLIED SEPARATELY.
C                            (DESIGNED FOR NEUTRAL STATE).
C  INPUT : (C*2)  YEAR     = REFERENCE  YEAR  (ABBREVIATED)  OF   INPUT
C                            MASTER CONDENSED COLL.-DIEL. COEFFTS. FILE.
C  INPUT : (L*4)  SELTAB  = .TRUE.  => PREPARE ELEMENT MASTER FILE
C                           .FALSE. => DO NOT PREPARE ELEM. MASTER FILE
C  INPUT : (L*4)  REPTAB  = .TRUE.  => REPLACE EXISTING ELEMENT MASTER
C                                      FILES.
C                         = .FALSE. => DO NOT REPLACE EXISTING ELEMENT
C                                      MASTER FILES.
C                           ('REPTAB' IS IGNORED IF 'SELTAB'=.FALSE.)
C
C  INPUT : (I*4)  NDZ      = NUMBER OF CHARGE STATES
C  INPUT : (I*4)  NDDEN    = MAXIMUM NUMBER OF INPUT DENSITIES
C  INPUT : (I*4)  NDTIN    = MAXIMUM NUMBER OF INPUT TEMPERATURES
C
C  INPUT : (I*4)  IZ0      = ELEMENT NUCLEAR CHARGE
C                                    (DETERMINES OUTPUT FILE NAME)
C  INPUT : (I*4)  IZ1      = MINIMUM ALLOWED IONIC CHARGE + 1
C                            (ACCORDING TO AVAILABLE NO. OF SEQUENCES
C                             STORED IN FILES FOR 'YEAR')
C  INPUT : (I*4)  NEL1     = NUMBER OF ELECTRONS IN STATE 'IZ1'
C  INPUT : (I*4)  IZ2      = MAXIMUM ALLOWED IONIC CHARGE + 1
C  INPUT : (I*4)  NEL2     = NUMBER OF ELECTRONS IN STATE 'IZ2'
C
C  INPUT : (R*8)  T()      = SET OF 'MAXT' ELECTRON TEMPERATURES: KELVIN
C  INPUT : (R*8)  TL()     = LOG10('T()')
C  INPUT : (I*4)  MAXT     = NUMBER OF TEMPERATURES (<= 'NDTIN' )
C
C  INPUT : (R*8)  DENSA()  = SET OF 'MAXD' ELECTRON DENSITIES: CM-3
C  INPUT : (R*8)  DENSL()  = LOG10 ('DENSA()')
C  INPUT : (I*4)  MAXD     = NUMBER OF DENSITIES (<= 'NDDEN' )
C
C  OUTPUT: (R*8)  ATTY(,) = WORKING SPACE FOR 3-WAY SPLINE ITERPOLATION
C                           ( STORES LOG10(INTERPOLATED VALUES) )
C                            1ST DIMENSION: TEMPERATURE
C                            2ND DIMENSION: DENSITY
C  OUTPUT: (R*8)  ARRAY(,,)= LOG10(INTERPOLATED DATA) FOR:
C                             1ST ARRAY DIMENSION - ION CHARGE/STAGE
C                             2ND ARRAY DIMENSION - TEMPERATURE
C                             3RD ARRAY DIMENSION - DENSITY
C  OUTPUT: (L*4)  ZINTRP() = .TRUE.  => 'ARRAY(,,)' VALUE FOR CHARGE-
C                                       STATE INTERPOLATED.
```

```
C                          = .FALSE. => 'ARRAY(,,)' VALUE FOR CHARGE-
C                                       STATE EXTRAPOLATED.
C                            1ST DIMENSION: CHARGE-STATE INDEX
C  OUTPUT: (L*4)  DINTRP(,)= .TRUE.  => 'ARRAY(,,)' VALUE FOR DENSITY
C                                       INDEX AND CHARGE INTERPOLATED.
C                          = .FALSE. => 'ARRAY(,,)' VALUE FOR DENSITY
C                                       INDEX AND CHARGE EXTRAPOLATED.
C                            1ST DIMENSION: DENSITY INDEX
C                            2ND DIMENSION: CHARGE-STATE INDEX
C  OUTPUT: (L*4)  TINTRP(,)= .TRUE.  =>'ARRAY(,,)' VALUE FOR TEMPERATURE
C                                       INDEX AND CHARGE INTERPOLATED.
C                          = .FALSE. =>'ARRAY(,,)' VALUE FOR TEMPERATURE
C                                       INDEX AND CHARGE EXTRAPOLATED.
C                            1ST DIMENSION: TEMPERATURE INDEX
C                            2ND DIMENSION: CHARGE-STATE INDEX
C  OUTPUT: (C*80) DSNO()  = OUPUT MASTER FILE NAME FOR EACH DATA TYPE
C
C  OUTPUT: (C*80) DSNIN(,)= INPUT FILE NAME FOR EACH DATA TYPE AND
C                           CHARGE
C
C          (I*4)  NKDIM   = PARAMETER =
C                           MAXIMUM  ARRAY  DIMENSIONS  FOR  CONDENSED
C                           MASTER FILE DATA FOR A GIVEN CHARGE STATE.
C          (I*4)  IUNT12  = PARAMETER = UNIT FOR READING DATA = 12
C
C          (R*8)  LOGMIN  = PARAMETER = MINIMUM LOG VALUE ALLOWED
C
C          (C*2)  XFESYM  = FUNCTION - (SEE ROUTINES SECTION BELOW)
C          (C*30) DSNAME  = INPUT MASTER CONDENSED FILE DATA SET NAME
C          (C*30) DSNOUT  = OUTPUT ELEMENT MASTER FILE DATA SET NAME
C          (C*2)  SEQUA   = ELEMENT SYMBOL FOR GIVEN NUCLEAR CHARGE
C          (C*3)  CDTYP() = INPUT MASTER CONDENSED FILE TYPE  USED  FOR
C                           CONSTRUCTING 'DSNAME'/'DSNOUT'. () = 'ISWIT'
C
C          (L*4)  LEXIST  = .TRUE.  => STANDARD MASTER CONDENSED FILE
C                                      EXISTS.
C                           .FALSE. => STANDARD MASTER CONDENSED FILE
C                                      DOES NOT EXIST.
C          (L*4)  LERROR  = .TRUE.  => ERROR FOUND IN READING STANDARD
C                                      MASTER CONDENSED FILE.
C                           .FALSE  =>NO ERROR FOUND IN READING STANDARD
C                                      MASTER CONDENSED FILE.
C          (L*4)  LSWIT   = .TRUE.  => SET OF 'EIA' VALUES PRESENT IN
C                                      MASTER CONDENSED FILE.
C                           .FALSE  => SET OF 'EIA' VALUES NOT PRESENT
C                                      IN MASTER CONDENSED FILE.
C
C          (I*4)  I4UNIT  = FUNCTION (SEE ROUTINE SECTION BELOW)
C          (I*4)  IZSTRT  = 'IZ1'
C          (I*4)  IZSTOP  = 'IZ2'
C          (I*4)  NZMAX   = 'NDZ'
C          (I*4)  NDMAX   = 'NDDEN'
C          (I*4)  NTMAX   = 'NDTIN'
C          (I*4)  ID      = ARRAY SUBSCRIPT USED FOR DENSITY VALUES
C          (I*4)  IS      = ARRAY SUBSCRIPT USED FOR SEQUENCE VALUES.
C                           REPRESENTS NUCLEAR CHARGE FOR ISO-ELECTRONIC
C                           SEQUENCE ELEMENT.
C                           (IMPLIES NUCLEAR CHARGE 'IS'-LIKE SEQUENCE)
C          (I*4)  IT      = ARRAY SUBSCRIPT USED FOR TEMPERATURE VALUES
C          (I*4)  IDE     = NUMBER OF REDUCED DENSITIES READ FROM INPUT
C                           MASTER CONDENSED FOR SEQUENCE 'IS'.
```

```
C          (I*4)  ITE     = NO. OF REDUCED TEMPERATURES READ FROM INPUT
C                            MASTER CONDENSED FOR SEQUENCE 'IS'.
C          (I*4)  IZE     = NO. OF CHARGE STATES GIVEN IN THE INPUT
C                            MASTER CONDENSED FOR SEQUENCE 'IS'.
C          (I*4)  IZF     = ELEMENT RECOMBINING ION CHARGE (IZ0+1-IS)
C          (I*4)  LS      = NON-BLANK LENGTH OF 'SEQUA'.
C          (I*4)  LD1     = VALUE  FOR  FIRST  DIMENSION OF 'ARRAY(,,)'
C                            (REPRESENTS STAGE/ION CHARGE)
C
C          (R*8)  ZIPT()  = SET OF 'IZE' INPUT RECOMBINING ION CHARGES
C                            READ FROM CONDENSED MASTER FILE.
C          (R*8)  TR()    = SET OF 'ITE'  INPUT  REDUCED  TEMPERATURES
C                            (K/Z1**2) READ FROM CONDENSED MASTER FILE.
C          (R*8)  DENSR() = SET OF 'IDE' INPUT REDUCED DENSITIES (CM-3/
C                            Z1**7) READ FROM CONDENSED MASTER FILE.
C          (R*8)  AIPT(,,)= CONDENSED MASTER FILE DATA. COLL-DIEL COEFF.
C                            1ST DIMENSION: REDUCED DENSITY ('DENSR()')
C                            2ND DIMENSION: REDUCED TEMPERATURE ('TR()')
C                            3RD DIMENSION: CHARGE STATE ('ZIPT()')
C          (R*8)  EIA()   = IONISATION RATE COEFFICIENTS: ()=ION CHARGE
C                            (UNITS: PRIOR TO 'XXCEIA' CALL: WAVE NUMBERS
C                                    AFTER CALL TO 'XXCEIA': RYDBERGS )
C NOTE:
C          STREAM HANDLING:
C                  STREAM 12 IS USED FOR READING CONDENSED MASTER FILES
C                  STREAM 13 IS USED FOR WRITING  ELEMENT  MASTER FILES
C
C          THIS SUBROUTINE IS A STRUCTURED AND AMENDED VERSION OF  THE
C          SUBROUTINE 'EIONST' WRITTEN BY H.P. SUMMERS, JET  (VERSION:
C          2 NOV 1989 / 1FEB 1990).
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          D4OUTF     ADAS      OUTPUT OF ELEMENT MASTER FILE
C          DXSPL1     ADAS      1ST PART OF 3-WAY SPLINE OF INPUT DATA
C          DXSPL2     ADAS      2ND PART OF 3-WAY SPLINE OF INPUT DATA
C          DXSPL3     ADAS      3RD PART OF 3-WAY SPLINE OF INPUT DATA
C          XXOPEN     ADAS      INQUIRE AND OPEN A DATA SET
C          XXINST     ADAS      FETCH   DATA   FROM   STANDARD  MASTER
C                               CONDENSED FILE.
C          XXCEIA     ADAS      CONVERT ION. RATE COEF. FROM WAVE NOS TO
C                               RYDBERGS AND EXTRAPOLATE MISSING VALUES
C          XFESYM     ADAS      CHARACTER*2 FUNCTION -
C                               FETCH ELEMENT SYMBOL FOR GIVEN NUCLEAR
C                               CHARGE
C          XXTERM     ADAS      TERMINATES PROGRAM WITH MESSAGE
C          I4UNIT     ADAS      FETCH UNIT NUMBER FOR OUTPUT OF MESSAGES
C
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 4569
C
C DATE:    08/10/90
C
C DATE:    26/10/90 - PE BRIDEN (JET/TESSELLA) - MINIMUM LOG10 VALUE
C                     ALLOWED WAS SET EQUAL TO 'LOGMIN'.  I.E. GIVES
C                     LOWER LIMIT FOR 'ARRAY(,,)'.
C
C UPDATE:  29/01/91 - PE BRIDEN - ADAS91 -INTRODUCED 'TINTRP' & 'DINTRP'
```

```
C                                       - AMENDED ARGUMENT LIST
C                                       - AMENDED ARGUMENT LISTS FOR
C                                         'D4SPL2' AND 'D4SPL3'.
C
C UPDATE:  30/01/91 - PE BRIDEN - ADAS91 -INTRODUCED 'ZINTRP'
C
C UPDATE:  13/02/91 - PE BRIDEN - ADAS91 - REPLACED XXESYM WITH XFESYM
C
C UPDATE:  21/02/91 - PE BRIDEN - ADAS91: INTRODUCED 'IZSTRT' , 'IZSTOP'
C                                 'NZMAX', 'NDMAX' AND 'NTMAX' TO  STOP
C                                 ICA MESSAGES BEING GENERATED.
C
C UPDATE:  05/03/91 - PE BRIDEN - ADAS91: ADDED CALL TO 'XXOPEN' BEFORE
C                                         AMENDED 'XXINST'.
C
C UPDATE:  20/03/91 - PE BRIDEN - ADAS91: MAJOR   CHANGES   TO   SPLINE
C                                         ROUTINES 'D4SPL?' -> 'DXSPL?'.
C                                         DENSA() ADDED TO ARGUMENT LIST
C
C UPDATE:  23/04/93 - PE BRIDEN - ADAS91: ADDED I4UNIT FUNCTION TO WRITE
C                                         STATEMENTS FOR SCREEN MESSAGES
C
C UPDATE:  24/05/93 - PE BRIDEN - ADAS91: CHANGED I4UNIT(0)-> I4UNIT(-1)
C
C UNIX-IDL PORT:
C
C VERSION: 1.1 DATE: 11-11-96
C MODIFIED: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C - FIRST CONVERTED
C
C VERSION: 1.2 DATE: 20-11-96
C MODIFIED: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C - TIDIED OUTPUT
C
C VERSION: 1.3 DATE: 28-11-96
C MODIFIED: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C - CORRECTED DEFINITION OF OPEN17 TO LOGICAL TYPE
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      CHARACTER*8         DATE
      CHARACTER*80        DSNIN(50,10),             DSNO(10)
      CHARACTER*32        TITLE
      CHARACTER*6         UIDIN,       USERID
      CHARACTER*2         YEAR
      INTEGER             ISWIT,       IZ0,         IZ1,         IZ2
      INTEGER             MAXD,        MAXT,        NDDEN,       NDTIN
      INTEGER             NDZ,         NEL1,        NEL2,        NIND
      LOGICAL             DINTRP(NDDEN,NDZ),        OPEN17,      REPTAB
      LOGICAL             SELTAB,      TINTRP(NDTIN,NDZ)
      LOGICAL             ZINTRP(NDZ)
      REAL*8              ARRAY(NDZ,NDTIN,NDDEN),   ATTY(NDTIN,NDDEN)
      REAL*8              DENSA(NDDEN),             DENSL(NDDEN)
      REAL*8              T(NDTIN),    TL(NDTIN)
```

5.12 d4ibal: Subroutine d4ibal from library adas4xx

```
      SUBROUTINE D4IBAL ( MAXT  , IZ1   , IZ2  ,
     &                    NDZ   , NDTIN ,
     &                    ACDL  , SCDL
     &                  )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: D4IBAL *********************
C
C  PURPOSE: TO CALCULATE IONISATION BALANCES AT FIXED DENSITY
C
C  CALLING PROGRAM: ADAS404
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  MAXT    = NUMBER OF USER ENTERED TEMPERATURES <= NDTIN
C  INPUT : (I*4)  IZ1     = MINIMUM ALLOWED IONIC CHARGE + 1
C                                    (ACCORDING TO AVAILABLE 'NSET')
C  INPUT : (I*4)  IZ2     = MAXIMUM ALLOWED IONIC CHARGE + 1
C
C  INPUT : (I*4)  NDZ     = NUMBER OF CHARGE STATES
C  INPUT : (I*4)  NDTIN   = MAXIMUM NUMBER OF INPUT TEMPERATURES
C
C  I/O   : (R*8)  ACDL(,) = INPUT : LOG10(RECOMB. COLL-DIEL COEFF)
C                           OUTPUT: LOG10(IONISATION-BALANCE)
C                           NOTE:   THESE VALUES ARE FOR A FIXED DENSITY
C                              1ST ARRAY DIMENSION = ION CHARGE/STAGE
C                              2ND ARRAY DIMENSION = TEMPERATURE
C  I/O   : (R*8)  SCDL(,) = INPUT : LOG10(IONIS. COLL-DIEL COEFF)
C                           OUTPUT: IONISATION-BALANCE
C                           NOTE:   THESE VALUES ARE FOR A FIXED DENSITY
C                              1ST ARRAY DIMENSION = ION CHARGE/STAGE
C                              2ND ARRAY DIMENSION = TEMPERATURE
C
C          (I*4)  MAX     = MAXIMUM NUMBER OF CHARGES FOR COLL-DIEL COEF
C          (I*4)  MAX1    = MAXIMUM NUMBER OF CHARGES FOR ION.-BALANCE
C                           ('MAX' + 1)
C          (I*4)  IZ      = ARRAY SUBSCRIPT USED FOR ION CHARGE VALUES
C          (I*4)  IT      = ARRAY SUBSCRIPT USED FOR TEMPERATURE VALUES
C          (I*4)  IREF    = REFERENCE POINT IN COLL-DIEL COEFF ARRAY,
C                           REPRESENTING THE CHARGE BELOW WHICH THE ION.
C                           COLL-DIEL COEFF IS GREATER THAN RECOMB COLL-
C                           DIEL. COEFF..
C
C          (R*8)  S       = USED FOR SUMMING COEFFICIENTS
C          (R*8)  SLOG    = LOG10( 'S' )
C
C
C ROUTINES: NONE
C
C AUTHOR:   PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C           K1/0/81
C           JET EXT. 4569
C
C DATE:     08/10/90
C
C UNIX-IDL PORT:
C
C VERSION: 1.1 DATE: 11-11-96
C MODIFIED: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
```

```
C - FIRST CONVERTED
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      INTEGER             IZ1,         IZ2,         MAXT,        NDTIN
      INTEGER             NDZ
      REAL*8              ACDL(NDZ,NDTIN),          SCDL(NDZ,NDTIN)
```

5.13 d4lbal: Subroutine d4lbal from library adas4xx

```
      SUBROUTINE D4LBAL( NDZ    ,          NDDEN , NDTIN ,
     &                   IZ1    , IZ2    , MAXD  , MAXT  ,
     &                   LDACDL , LDSCDL ,
     &                   LTACDL , LTSCDL ,
     &                   LDIBAL ,
     &                   LTIBAL
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: D4LBAL *********************
C
C  PURPOSE:  TO IDENTIFY THE TEMPERATURES AND DENSITY FOR WHICH THE
C            CALCULATION OF IONISATION BALANCES INVOLVED THE USE OF
C            EXTRAPOLATED DATA.
C
C  CALLING PROGRAM: ADAS404
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  NDZ     = NUMBER OF CHARGE STATES
C  INPUT : (I*4)  NDDEN   = MAXIMUM NUMBER OF INPUT DENSITIES
C  INPUT : (I*4)  NDTIN   = MAXIMUM NUMBER OF INPUT TEMPERATURES
C
C  INPUT : (I*4)  IZ1     = MINIMUM ALLOWED IONIC CHARGE + 1
C  INPUT : (I*4)  IZ2     = MAXIMUM ALLOWED IONIC CHARGE + 1
C  INPUT : (I*4)  MAXD    = NUMBER OF USER ENTERED DENSITIES (<=NDDEN)
C  INPUT : (I*4)  MAXT    = NUMBER OF USER ENTERED TEMPERATURES(<=NDTIN)
C
C  INPUT : (L*4)  LDACDL(,)=.TRUE. => RECOMBINATION COLL.-DIEL COEFFT.
C                                     VALUE FOR DENSITY     INDEX AND
C                                     CHARGE INTERPOLATED.
C                           =.FALSE.=> RECOMBINATION COLL.-DIEL COEFFT.
C                                     VALUE FOR DENSITY     INDEX AND
C                                     CHARGE EXTRAPOLATED.
C                            1ST DIMENSION: DENSITY     INDEX
C                            2ND DIMENSION: CHARGE-STATE INDEX
C  INPUT : (L*4)  LDSCDL(,)=.TRUE. => IONIZATION    COLL.-DIEL COEFFT.
C                                     VALUE FOR DENSITY     INDEX AND
C                                     CHARGE INTERPOLATED.
C                           =.FALSE.=> IONIZATION    COLL.-DIEL COEFFT.
C                                     VALUE FOR DENSITY     INDEX AND
C                                     CHARGE EXTRAPOLATED.
C                            1ST DIMENSION: DENSITY     INDEX
C                            2ND DIMENSION: CHARGE-STATE INDEX
C
C  INPUT : (L*4)  LTACDL(,)=.TRUE. => RECOMBINATION COLL.-DIEL COEFFT.
C                                     VALUE FOR TEMPERATURE INDEX AND
C                                     CHARGE INTERPOLATED.
C                           =.FALSE.=> RECOMBINATION COLL.-DIEL COEFFT.
C                                     VALUE FOR TEMPERATURE INDEX AND
C                                     CHARGE EXTRAPOLATED.
C                            1ST DIMENSION: TEMPERATURE INDEX
C                            2ND DIMENSION: CHARGE-STATE INDEX
C  INPUT : (L*4)  LTSCDL(,)=.TRUE. => IONIZATION    COLL.-DIEL COEFFT.
C                                     VALUE FOR TEMPERATURE INDEX AND
C                                     CHARGE INTERPOLATED.
C                           =.FALSE.=> IONIZATION    COLL.-DIEL COEFFT.
C                                     VALUE FOR TEMPERATURE INDEX AND
C                                     CHARGE EXTRAPOLATED.
```

```
C                           1ST DIMENSION: TEMPERATURE INDEX
C                           2ND DIMENSION: CHARGE-STATE INDEX
C
C  OUTPUT: (L*4)  LDIBAL() =.TRUE. => IONIZATION BALANCES FOR DENSITY
C                                     INVOLVE NOT EXTRAPOLATION.
C                            =.FALSE.=> IONIZATION BALANCES FOR DENSITY
C                                     INVOLVE EXTRAPOLATION  IN  SOME
C                                     PART OF THEIR CALCULTION.
C                            1ST DIMENSION: DENSITY INDEX
C  OUTPUT: (L*4)  LTIBAL() =.TRUE. => IONIZATION BALANCES FOR TEMP'TURE
C                                     INVOLVE NOT EXTRAPOLATION.
C                            =.FALSE.=> IONIZATION BALANCES FOR TEMP'TURE
C                                     INVOLVE EXTRAPOLATION  IN  SOME
C                                     PART OF THEIR CALCULTION.
C                            1ST DIMENSION: TEMPERATURE INDEX
C
C          (I*4)  NZ      = NUMBER OF CHARGE STATES TO BE ASSESSED
C          (I*4)  ID      = ARRAY SUBSCRIPT USED FOR DENSITY INDEX
C          (I*4)  IT      = ARRAY SUBSCRIPT USED FOR TEMPERATURE INDEX
C          (I*4)  IZ      = ARRAY SUBSCRIPT USED FOR CHARGE-STATE INDEX
C
C          (L*4)  LS      = .TRUE.  => NO EXTRAPOLATION
C                           .FALSE. => EXTRAPOLATION
C
C NOTE:
C
C ROUTINES: NONE
C
C AUTHOR:   PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C           K1/0/81
C           JET EXT. 4569
C
C DATE:     30/01/91 - ADAS91 ROUTINE
C
C UNIX-IDL PORT:
C
C VERSION: 1.1 DATE: 11-11-96
C MODIFIED: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C - FIRST CONVERTED
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IZ1,         IZ2,         MAXD,        MAXT
      INTEGER             NDDEN,       NDTIN,       NDZ
      LOGICAL             LDACDL(NDDEN,NDZ),        LDIBAL(NDDEN)
      LOGICAL             LDSCDL(NDDEN,NDZ),        LTACDL(NDTIN,NDZ)
      LOGICAL             LTIBAL(NDTIN),            LTSCDL(NDTIN,NDZ)
```

5.14 d4open: Subroutine d4open from library adas4xx

```
       SUBROUTINE D4OPEN( IUNIT, DSFULL , LEXIST )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: D4OPEN *********************
C
C  PURPOSE: TO INQUIRE & OPEN INPUT DATA FILE & ALLOCATE TO UNIT 'IUNIT'
C           (READ ONLY)  - IF IT DOES NOT EXISTS A MESSAGE IS  SENT  TO
C           THE SCREEN AND LEXIST IS RETURNED AS FALSE.
C
C           *** THIS VERSION SUPPRESSES ERROR MESSAGE ON NON-EXISTING
C               FILES. IT IS OTHERWISE IDENTICAL TO XXOPEN.
C
C  CALLING PROGRAM:  ADAS404
C
C  SUBROUTINE:
C
C  INPUT :    (I*4)  IUNIT   = UNIT TO WHICH INPUT FILE IS ALLOCATED
C  INPUT :    (C*(*))DSFULL  = FULL INPUT DATA SET NAME (INCL. USERID)
C                              IN FORM FOR DYNAMIC ALLOCATION.
C  OUTPUT:    (L*4)  LEXIST  = .TRUE.  => DATA SETS EXISTS AND IS OPEN
C                            = .FALSE. => DATA SET DOES NOT EXIST
C
C             (I*4)  I4UNIT  = FUNCTION (SEE ROUTINE SECTION BELOW)
C             (C*1)  BSLASH  = '/' - MUST BE FIRST 'DSFULL'CHARACTER
C NOTE:
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4UNIT     ADAS      FETCH UNIT NUMBER FOR OUTPUT OF MESSAGES
C          XXTERM     ADAS      TERMINATES ADAS PROGRAM WITH MESSAGE
C
C AUTHOR:  LORNE HORTON (JET)
C      - BASED ON XXOPEN.
C
C DATE:    20-10-97
C-----------------------------------------------------------------------
C VERSION: 1.1 DATE:27-02-98
C MODIFIED: RICHARD MARTIN
C - PUT UNDER SCCS CONTROL.
C
C VERSION: 1.2 DATE:17-05-07
C MODIFIED: Allan Whiteford
C - Updated comments as part of subroutine documentation
C                 procedure.
C-----------------------------------------------------------------------
      CHARACTER*(*)       DSFULL
      INTEGER             IUNIT
      LOGICAL             LEXIST
```

5.15 d4spln: Subroutine d4spln from library adas4xx

```
      SUBROUTINE D4SPLN( ISWIT  , LSWIT  ,
     &                   NUDMAX , NUTMAX ,
     &                   NDDEN  , NDTIN  , NDZ1V  ,
     &                   MAXT   , MAXD   ,
     &                   IDE    , ITE    , IZE    ,
     &                   DUSR   , TUSR   , IZ1    ,
     &                   DENSR  , TR     , ZIPT   ,
     &                   EIA    , AIPT   ,
     &                   LZRNG  , LDRNG  , LTRNG  ,
     &                   ATTY   , AOUT
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: D4SPLN *********************
C
C  PURPOSE: TO INTERPOLATE/EXTRAPOLATED DATA FROM MASTER CONDENSED  FILE
C           TO THE USER ENTERED TEMPERATURE/DENSITY ARRAY FOR THE SELEC-
C           TED RECOMBINING ION CHARGE.
C           BASED ON ADAS9140.FORT(D1SPLN)
C
C  CALLING PROGRAM: LH404RU
C
C  DATA:
C
C           THE  SOURCE  DATA IS CONTAINED AS  MEMBERS  OF  PARTITIONED
C           DATA SETS AS FOLLOWS:
C
C             1. JETUID.ACD<YR>.DATA
C             2. JETUID.SCD<YR>.DATA
C             3. JETUID.CCD<YR>.DATA
C             4. JETUID.PRB<YR>.DATA
C             5. JETUID.PRC<YR>.DATA
C             6. JETUID.QCD<YR>.DATA
C             7. JETUID.XCD<YR>.DATA
C             8. JETUID.PLT<YR>.DATA
C             9. JETUID.PLS<YR>.DATA
C            10. JETUID.MET<YR>.DATA
C
C
C           WHERE <YR> DENOTES TWO INTEGERS FOR THE YEAR SELECTED.
C
C           THE PARTICULAR TYPE OPENED (1-10) IS SELECTED BY 'ISWIT'
C
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  ISWIT   = DATA TYPE SELECTOR (SEE ABOVE) (1 -> 10)
C  INPUT : (L*4)  LSWIT   = .TRUE.  => SET OF 'EIA' VALUES PRESENT IN
C                                      MASTER CONDENSED FILE.
C                           .FALSE  => SET OF 'EIA' VALUES NOT PRESENT
C                                      IN MASTER CONDENSED FILE.
C
C  INPUT : (I*4)  NUDMAX  = USER ENTERED VALUES -
C                           MAXIMUM NUMBER OF DENSITY VALUES
C  INPUT : (I*4)  NUTMAX  = USER ENTERED VALUES -
C                           MAXIMUM NUMBER OF TEMPERATURE VALUES
C
C  INPUT : (I*4)  NDDEN   = INPUT MASTER CONDENSED FILE -
C                           MAXIMUM NUMBER OF REDUCED DENSITIES
```

```
C  INPUT : (I*4)  NDTIN   = INPUT MASTER CONDENSED FILE -
C                           MAXIMUM NUMBER OF REDUCED TEMPERATURES
C  INPUT : (I*4)  NDZ1V   = INPUT MASTER CONDENSED FILE -
C                           MAXIMUM NUMBER OF CHARGE STATES
C
C  INPUT : (I*4)  MAXT    = USER ENTERED VALUES -
C                           NUMBER OF TEMPERATURE VALUES ENTERED
C  INPUT : (I*4)  MAXD    = USER ENTERED VALUES -
C                           NUMBER OF DENSITY VALUES ENTERED
C
C  INPUT : (I*4)  IDE     = INPUT MASTER CONDENSED FILE -
C                           NUMBER OF REDUCED DENSITIES READ
C  INPUT : (I*4)  ITE     = INPUT MASTER CONDENSED FILE -
C                           NUMBER OF REDUCED TEMPERATURES READ
C  INPUT : (I*4)  IZE     = INPUT MASTER CONDENSED FILE -
C                           NUMBER OF CHARGE STATES READ
C
C  INPUT : (R*8)  DUSR()  = USER ENTERED VALUES -
C                           SET OF 'IUDVAL' ENTERED ELECTRON DENSITIES
C                           (UNITS: CM**-3)
C                           DIMENSION: TEMPERATURE/DENSITY PAIR INDEX
C  INPUT : (R*8)  TUSR()  = USER ENTERED VALUES -
C                           SET OF'IUTVAL' ENTERED ELECTRON TEMPERATURES
C                           (UNITS: KELVIN)
C                           DIMENSION: TEMPERATURE/DENSITY PAIR INDEX
C  INPUT : (I*4)  IZ1     = USER ENTERED VALUE -
C                           RECOMBINING ION CHARGE
C
C  INPUT : (R*8)  DENSR() = INPUT MASTER CONDENSED FILE -
C                           SET OF 'IDE' REDUCED DENSITIES (CM-3/Z1**7)
C  INPUT : (R*8)  TR()    = INPUT MASTER CONDENSED FILE -
C                           SET OF 'ITE' REDUCED TEMPERATURES (K/Z1**2)
C  INPUT : (R*8)  ZIPT()  = INPUT MASTER CONDENSED FILE -
C                           SET OF 'IZE' INPUT RECOMBINING ION CHARGES
C
C  INPUT : (R*8)  EIA()   = INPUT MASTER CONDENSED FILE -
C                           IONISATION RATE COEFFTS. - (UNITS: RYDBERGS)
C                           DIMENSION: ION CHARGE
C  INPUT : (R*8)  AIPT(,,)= INPUT MASTER CONDENSED FILE -
C                           RELEVANT COEFFICIENT/POWER DATA FOR 'ISWIT'.
C                           1ST DIMENSION: DENSITY INDEX     ('DENSR()')
C                           2ND DIMENSION: TEMPERATURE INDEX  ('TR()')
C                           3RD DIMENSION: CHARGE STATE INDEX ('ZIPT()')
C
C  OUTPUT: (L*4)  LZRNG(1) = .TRUE.  => 'AOUT()' VALUES FOR CHARGE-
C                                       STATE 'IZ1' INTERPOLATED.
C                          = .FALSE. => 'AOUT()' VALUE  FOR CHARGE-
C                                       STATE 'IZ1' EXTRAPOLATED.
C  OUTPUT: (L*4)  LDRNG()  = .TRUE.  => 'AOUT()' VALUE FOR DENSITY
C                                       INDEX INTERPOLATED.
C                          = .FALSE. => 'AOUT()' VALUE FOR DENSITY
C                                       INDEX EXTRAPOLATED.
C                           DIMENSION: DENSITY INDEX
C  OUTPUT: (L*4)  LTRNG()  = .TRUE.  => 'AOUT()' VALUE FOR TEMPERATURE
C                                       INDEX INTERPOLATED.
C                          = .FALSE. => 'AOUT()' VALUE FOR TEMPERATURE
C                                       INDEX EXTRAPOLATED.
C                           DIMENSION: TEMPERATURE INDEX
C
C  OUTPUT: (R*8)  AOUT(,)  = EXTRAPOLATED/INTERPOLATED  DATA  FOR
C                            USER ENTERED TEMPERATURE/DENSITY ARRAY.
```

```
C                           ( STORES LOG10(INTERPOLATED VALUES) )
C                              1ST DIMENSION: TEMPERATURE
C                              2ND DIMENSION: DENSITY
C
C          (I*4)  NUDIM   = PARAMETER = MUST BE GREATER THAN OR EQUAL TO
C                                       'NUDMAX' AND 'NUTMAX'
C
C          (I*4)  NDMAX1  = 'NDDEN'
C          (I*4)  NTMAX1  = 'NDTIN'
C          (I*4)  NZMAX1  = 'NDZ1V'
C          (I*4)  NDMAX2  = 'NUDMAX'
C          (I*4)  NTMAX2  = 'NUTMAX'
C          (I*4)  ITD     = GENERAL USE ARRAY SUBSCRIPT INDEX
C          (I*4)  IDD     = GENERAL USE ARRAY SUBSCRIPT INDEX
C
C          (R*8)  ATTY(,) = WORKING SPACE FOR 3-WAY SPLINE ITERPOLATION
C                           ( STORES LOG10(INTERPOLATED VALUES) )
C                            1ST DIMENSION: TEMPERATURE
C                            2ND DIMENSION: DENSITY
C
C  PARAMETER (I*4)  DLOGMIN = SETS MINIMUM VALUE OF LOG OF COEFFICIENT
C
C NOTE:
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          XUFLOW
C          DXSPL1     ADAS      1ST PART OF 3-WAY SPLINE OF INPUT DATA
C          DXSPL2     ADAS      2ND PART OF 3-WAY SPLINE OF INPUT DATA
C          DXSPL3     ADAS      3RD PART OF 3-WAY SPLINE OF INPUT DATA
C
C
C AUTHOR:  WILLIAM J. DICKSON       12/12/92
C          (REFER TO DOCUMENTATION FOR   D1SPLN)
C
C UPDATES FROM D1SPLN:
C
C   12/12/92   ARRAY BOUNDS FOR ATTY SET EQUAL TO THOSE IN DXSPL1 ETC
C              THERFORE INCLUDE ATTY IN CALL LIST
C   22/02/96   HOUSECLEANING AFTER COPY FOR USE WITH LH404RU
C-----------------------------------------------------------------------
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 11-11-96
C MODIFIED: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST CONVERTED
C
C
C VERSION : 1.2
C DATE    : 23-02-2006
C MODIFIED: Martin O'Mullane
C           - Remove mainframe listing information in columns 72-80.
C
C-----------------------------------------------------------------------
      INTEGER             IDE,         ISWIT,       ITE,         IZ1
      INTEGER             IZE,         MAXD,        MAXT,        NDDEN
      INTEGER             NDTIN,       NDZ1V,       NUDMAX,      NUTMAX
      LOGICAL             LDRNG(NUDMAX),            LSWIT
      LOGICAL             LTRNG(NUTMAX),            LZRNG(1)
      REAL*8              AIPT(NDDEN,NDTIN,NDZ1V),  AOUT(NUTMAX,NUDMAX)
```

```
REAL*8              ATTY(NUTMAX,NUDMAX),      DENSR(NDDEN)
REAL*8              DUSR(NUDMAX),             EIA(50)
REAL*8              TR(NDTIN),   TUSR(NUTMAX)
REAL*8              ZIPT(NDZ1V)
```

5.16 d4tlog: Subroutine d4tlog from library adas4xx

```
      SUBROUTINE D4TLOG( INTYP, ITVAL, TEMP )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: D4TLOG *********************
C
C  PURPOSE:      TO CONVERT AN ARRAY OF LOG10(TEMPERATURES) FROM:
C           (KELVIN TO ELECTRON VOLTS) OR (ELECTRON VOLTS TO KELVIN)
C
C  CALLING PROGRAM:  D4OUTF
C
C  SUBROUTINE:
C
C  INPUT :    (I*4)  INTYP   = 1 => INPUT 'TEMP(array)' UNITS: KELVIN
C                            = 2 => INPUT 'TEMP(array)' UNITS: eV
C  INPUT :    (I*4)  ITVAL   = NUMBER OF TEMPERATURES IN 'TIN(array)'
C  I/O   :    (R*8)  TEMP()  = LOG10(INPUT TEMPERATURES (STATED UNITS))
C
C             (R*8)  EV2KEL  = LOG10( ELEC.VOLTS TO KELVIN CONVERSION)
C             (R*8)  TCONV   = TEMPERATURE CONVERSION PARAMETER
C
C ROUTINES:  NONE
C
C NOTE:
C            TEMPERATURE CONVERSION PARAMETERS:
C
C            INTYP = 1 ; TCONV =>             KELVIN  -> eV
C            INTYP = 2 ; TCONV =>                 eV  -> KELVIN
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 4569
C
C DATE:    08/10/90
C
C UNIX-IDL PORT:
C
C VERSION: 1.1 DATE: 11-11-96
C MODIFIED: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C - FIRST CONVERTED
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             INTYP,       ITVAL
      REAL*8              TEMP(ITVAL)
```

5.17 d4wmet: Subroutine d4wmet from library adas4xx

```
C
      SUBROUTINE D4WMET( IUNIT  , IZ0 )

C-----------------------------------------------------------------------
C
C  ******************** FORTRAN77 SUBROUTINE: D4WMET ******************
C
C  PURPOSE:  Write the number of metastables for each ionisation
C            stage to the top of resolved adf11 files.
C
C
C  CALLING PROGRAM: ADAS404 - lh404rr
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  IUNIT    = OUTPUT UNIT NUMBER
C  INPUT : (I*4)  IZ0      =         NUCLEAR CHARGE READ
C
C
C  ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C
C
C AUTHOR:  Martin O'Mullane
C
C DATE:    4/11/99
C
C VERSION:  1.1                                    DATE: 4/11/99
C MODIFIED: Martin O'Mullane
C           - First version
C
C-----------------------------------------------------------------------
      INTEGER             IUNIT,       IZ0
```

5.18 d4znel: Subroutine d4znel from library adas4xx

```
      SUBROUTINE D4ZNEL ( IZ1  , IZ2  ,
     &                    NEL1 , NEL2 ,
     &                    IZ0  , IZE1 , IZE2
     &                  )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: D4ZNEL *********************
C
C  PURPOSE: TO RETURN THE MAXIMUM AND MINIMUM CHOSEN FOR THE IONIC
C           CHARGE (+1) AND THE NUMBER OF ELECTRONS IN EACH CASE.
C
C  CALLING PROGRAM: ADAS404
C
C  SUBROUTINE:
C
C  OUTPUT: (I*4)  IZ1     = MINIMUM ALLOWED IONIC CHARGE + 1
C  OUTPUT: (I*4)  IZ2     = MAXIMUM ALLOWED IONIC CHARGE + 1
C
C  OUTPUT: (I*4)  NEL1    = NUMBER OF ELECTRONS IN STATE 'IZ1'
C  OUTPUT: (I*4)  NEL2    = NUMBER OF ELECTRONS IN STATE 'IZ2'
C
C  INPUT : (I*4)  IZ0     = ELEMENT NUCLEAR CHARGE
C  INPUT : (I*4)  IZE1    = LOWEST ION CHARGE
C  INPUT : (I*4)  IZE2    = HIGHEST ION CHARGE  (NB. EXCLUDING THE  BARE
C                           NUCLEUS - ONE MORE STAGE IS BROUGHT IN AUTO-
C                           MATICALLY IN THE IONISATION BALANCE)
C
C  ROUTINES: NONE
C
C  AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C           K1/0/81
C           JET EXT. 4569
C
C  DATE:    08/10/90
C
C UNIX-IDL PORT:
C
C VERSION: 1.1 DATE: 11-20-96
C MODIFIED: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C - FIRST CONVERTED
C
C VERSION: 1.2 DATE: 20-10-97
C MODIFIED: LORNE HORTON (JET)
C       - REMOVED NONSENSE WITH NIND, NSET AND YEAR
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      INTEGER             IZ0,         IZ1,         IZ2,         IZE1
      INTEGER             IZE2,        NEL1,        NEL2
```

5.19 d5data: Subroutine d5data from library adas4xx

```
       SUBROUTINE D5DATA( DSFLLA , LSELA  , LEXSA  , LDEFA , LPART  ,
     &                    IZ0    , IZ1MIN , IZ1MAX , NPART  ,
     &                    NTDIM  , ITMAX  ,
     &                    ISDIMD , IZDIMD , ITDIMD , IPDIMD , NPARTR,
     &                    DTEV   , DDENS  ,
     &                    DTEVD  , DDENSD , DRCOFD , ZDATA  ,
     &                    DRCOFI ,
     &                    ACDA   , LACDA  ,
     &                    SCDA   , LSCDA  ,
     &                    CCDA   , LCCDA  ,
     &                    PRBA   , LPRBA  ,
     &                    PRCA   , LPRCA  ,
     &                    QCDA   , LQCDA  ,
     &                    XCDA   , LXCDA  ,
     &                    PLTA   , LPLTA
     &                  )
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: D5DATA *********************
C
C PURPOSE : TO EXTRACT A COMPLETE SET OF COLLISIONAL DIELECTRONIC DATA
C           FOR A TEMP/DENSITY MODEL
C           FROM EITHER PARTIAL (METASTABLE/PARENT RESOLVED) OR STANDARD
C           (UNRESOLVED) ISONUCLEAR MASTER FILES
C
C NOTE    : THE SOURCE DATA IS CONTAINED AS SEQUENTIAL DATASETS
C           WITH THE FOLLOWING NAMING CONVENTIONS:
C
C                   (1) JETSHP.ACD<YR>#<EL).<CODE>DATA
C                   (2) JETSHP.SCD<YR>#<EL>.<CODE>DATA
C                   (3) JETSHP.CCD<YR>#<EL>.<CODE>DATA
C                   (4) JETSHP.PRB<YR>#<EL>.<FILT>.<CODE>DATA
C                   (5) JETSHP.PRC<YR>#<EL>.<FILT>.<CODE>DATA
C                   (6) JETSHP.QCD<YR>#<EL>.<CODE>DATA
C                   (7) JETSHP.XCD<YR>#<EL>.<CODE>DATA
C                   (8) JETSHP.PLT<YR>#<EL>.<CODE>DATA
C
C       WHERE, <YR>   = TWO DIGIT YEAR NUMBER
C              <EL>   = ONE OR TWO CHARACTER ELEMENT SYMBOL
C              <CODE> = R       => PARTIAL DATA
C                       U       => PARTIAL DATA
C                       OMITTED => STANDARD DATA
C              <FILT> = SIX CHARACTER POWER FILTER CODE
C
C       AND DATA OF CLASSES 6 AND 7 DO NOT EXIST FOR THE STANDARD CASE.
C
C
C INPUT  : (C*120)DSFLLA()  = MASTER FILE DATA SET NAMES (FULL MVS DSN)
C                             (IN FORM SUITABLE FOR DYNAMIC ALLOCATION)
C INPUT  : (L*4)  LSELA()   = .TRUE.  => INPUT DATA SET TYPE FOR THIS
C                                        INDEX SELECTED
C                           = .FALSE. => INPUT DATA SET FOR THIS INDEX
C                                        NOT SELECTED
C INPUT  : (L*4)  LEXSA()   = .TRUE.  => INPUT DATA SET TYPE FOR THIS
C                                        SELECTED INDEX EXISTS
C                           = .FALSE. => INPUT DATA SET DOES NOT EXIST
C                                        FOR THIS SELECTED INDEX
C INPUT  : (L*4)  LDEFA()   = .TRUE.  => INPUT DATA SET TYPE FOR THIS
```

```
C                                          DEFAULT YEAR INDEX EXISTS
C                              = .FALSE. => INPUT DATA SET DOES NOT EXIST
C                                          FOR THIS DEFAULT YEAR INDEX
C INPUT  : (L*4)  LPART      = .TRUE.  => PARTIAL DATA SELECTED
C                              = .FALSE. => STANDARD DATA SELECTED
C INPUT  : (I*4)  IZ0        = NUCLEAR CHARGE
C INPUT  : (I*4)  IZ1MIN     = MINIMUM ION CHARGE+1 IN MASTER DATA FILES
C INPUT  : (I*4)  IZ1MAX     = MAXIMUM ION CHARGE+1 IN MASTER DATA FILES
C INPUT  : (I*4)  NPART()    = METASTABLE PARTITION.  I.E. NUMBER OF
C                              METASTABLES FROM CHARGE STATE IZ1MIN-1 TO
C                              IZ1MAX ON INPUT
C INPUT  : (I*4)  NTDIM      = MAXIMUM NUMBER OF DTEV/DDENS PAIRS
C INPUT  : (I*4)  ITMAX      = NUMBER OF ( DTEV() , DDENS() ) PAIRS
C INPUT  : (I*4)  ISDIMD     = MAXIMUM NUMBER OF (CHARGE, PARENT, GROUND)
C                              BLOCKS IN ISONUCLEAR MASTER FILES
C INPUT  : (I*4)  IZDIMD     = MAXIMUM NUMBER OF CHARGE STATES
C                              IN ISONUCLEAR MASTER FILES
C INPUT  : (I*4)  ITDIMD     = MAXIMUM NUMBER OF TEMP OR DENS VALUES IN
C                              ISOELECTRONIC MASTER FILES
C INPUT  : (I*4)  IPDIMD     = MAXIMUM NUMBER OF METASTABLES FOR EACH
C                              IONISATION STAGE
C INPUT  : (R*8)  DTEV()     = DLOG10(ELECTRON TEMPERATURES (EV))
C INPUT  : (R*8)  DDENS()    = DLOG10(ELECTRON DENSITIES (CM-3))
C
C OUTPUT : (I*4)  NPARTR()   = METASTABLE PARTITION.  I.E. NUMBER OF
C                              METASTABLES FROM CHARGE STATE IZ1MIN-1 TO
C                              IZ1MAX FOUND IN MASTER FILE
C OUTPUT : (R*8)  DTEVD()    = DLOG10(DATA ELECTRON TEMPERATURES (EV))
C                              IN SELECTED MASTER FILE
C OUTPUT : (R*8)  DDENSD()   = DLOG10(DATA ELECTRON DENSITIES (CM-3))
C                              IN SELECTED MASTER FILE
C OUTPUT : (R*8)  DRCOFD(,,)= DLOG10(DATA RATE COEFFICIENTS (CM-3/S))
C                              IN SELECTED MASTER FILE
C                              1ST DIM: (CHARGE,META,GRD) BLOCK INDEX
C                              2ND DIM: TEMPERATURE INDEX
C                              3RD DIM: DENSITY INDEX
C OUTPUT : (R*8)  ZDATA()    = CHARGE + 1 FOR IONS IN SELECTED MASTER
C                              FILE
C                              1ST DIM: (CHARGE,META,GRD) BLOCK INDEX
C OUTPUT : (R*8)  DRCOFI()   = INTERPOLATION OF DRCOFD(,,) FOR
C                              DTEV() & DDENS()
C OUTPUT : (R*8)  ACDA(,,,) = INTERPOLATION OF ACD COEFFICIENT (CM3 S-1)
C                              1ST DIM: TEMPERATURE INDEX
C                              2ND DIM: CHARGE STATE INDEX
C                              3RD DIM: RECOMBINING METASTABLE INDEX
C                              4TH DIM: RECOMBINED METASTABLE INDEX
C OUTPUT : (L*4)  LACDA(,,) = .TRUE.  => ACD COEFFICIENT AVAILABLE
C                              .FALSE. => ACD COEFFICIENT NOT AVAILABLE
C                              1ST DIM: CHARGE STATE INDEX
C                              2ND DIM: RECOMBINING METASTABLE INDEX
C                              3RD DIM: RECOMBINED METASTABLE INDEX
C OUTPUT : (R*8)  SCDA(,,,) = INTERPOLATION OF SCD COEFFICIENT (CM3 S-1)
C                              1ST DIM: TEMPERATURE INDEX
C                              2ND DIM: CHARGE STATE INDEX
C                              3RD DIM: RECOMBINING METASTABLE INDEX
C                              4TH DIM: RECOMBINED METASTABLE INDEX
C OUTPUT : (L*4)  LSCDA(,,) = .TRUE.  => SCD COEFFICIENT AVAILABLE
C                              .FALSE. => SCD COEFFICIENT NOT AVAILABLE
C                              1ST DIM: CHARGE STATE INDEX
C                              2ND DIM: RECOMBINING METASTABLE INDEX
C                              3RD DIM: RECOMBINED METASTABLE INDEX
```

```
C OUTPUT : (R*8)  CCDA(,,,) = INTERPOLATION OF CCD COEFFICIENT (CM3 S-1)
C                             1ST DIM: TEMPERATURE INDEX
C                             2ND DIM: CHARGE STATE INDEX
C                             3RD DIM: RECOMBINING METASTABLE INDEX
C                             4TH DIM: RECOMBINED METASTABLE INDEX
C OUTPUT : (L*4)  LCCDA(,,) = .TRUE.  => CCD COEFFICIENT AVAILABLE
C                             .FALSE. => CCD COEFFICIENT NOT AVAILABLE
C                             1ST DIM: CHARGE STATE INDEX
C                             2ND DIM: RECOMBINING METASTABLE INDEX
C                             3RD DIM: RECOMBINED METASTABLE INDEX
C OUTPUT : (R*8)  PRBA(,,)  = INTERPOLATION OF PRB COEFFICIENT (W CM3 )
C                             1ST DIM: TEMPERATURE INDEX
C                             2ND DIM: CHARGE STATE INDEX
C                             3RD DIM: RECOMBINING METASTABLE INDEX
C OUTPUT : (L*4)  LCCDA(,)  = .TRUE.  => PRB COEFFICIENT AVAILABLE
C                             .FALSE. => PRB COEFFICIENT NOT AVAILABLE
C                             1ST DIM: CHARGE STATE INDEX
C                             2ND DIM: RECOMBINING METASTABLE INDEX
C OUTPUT : (R*8)  PRCA(,,)  = INTERPOLATION OF PRC COEFFICIENT (W CM3 )
C                             1ST DIM: TEMPERATURE INDEX
C                             2ND DIM: CHARGE STATE INDEX
C                             3RD DIM: RECOMBINING METASTABLE INDEX
C OUTPUT : (L*4)  LPRCA(,)  = .TRUE.  => PRC COEFFICIENT AVAILABLE
C                             .FALSE. => PRC COEFFICIENT NOT AVAILABLE
C                             1ST DIM: CHARGE STATE INDEX
C                             2ND DIM: RECOMBINING METASTABLE INDEX
C OUTPUT : (R*8)  QCDA(,,,) = INTERPOLATION OF QCD COEFFICIENT (CM3 S-1)
C                             1ST DIM: TEMPERATURE INDEX
C                             2ND DIM: CHARGE STATE INDEX
C                             3RD DIM: FIRST METASTABLE INDEX
C                             4TH DIM: SECOND METASTABLE INDEX
C OUTPUT : (L*4)  LQCDA(,,) = .TRUE.  => QCD COEFFICIENT AVAILABLE
C                             .FALSE. => QDC COEFFICIENT NOT AVAILABLE
C                             1ST DIM: CHARGE STATE INDEX
C                             2ND DIM: FIRST METASTABLE INDEX
C                             3RD DIM: SECOND METASTABLE INDEX
C OUTPUT : (R*8)  XCDA(,,,) = INTERPOLATION OF XCD COEFFICIENT (CM3 S-1)
C                             1ST DIM: TEMPERATURE INDEX
C                             2ND DIM: CHARGE STATE INDEX
C                             3RD DIM: FIRST PARENT METASTABLE INDEX
C                             4TH DIM: SECOND PARENT METASTABLE INDEX
C OUTPUT : (L*4)  LXCDA(,,) = .TRUE.  => XCD COEFFICIENT AVAILABLE
C                             .FALSE. => XDC COEFFICIENT NOT AVAILABLE
C                             1ST DIM: CHARGE STATE INDEX
C                             2ND DIM: FIRST PARENT METASTABLE INDEX
C                             3RD DIM: SECOND PARENT METASTABLE INDEX
C OUTPUT : (R*8)  PLTA(,,)  = INTERPOLATION OF PLT COEFFICIENT (W CM3 )
C                             1ST DIM: TEMPERATURE INDEX
C                             2ND DIM: CHARGE STATE INDEX
C                             3RD DIM: METASTABLE INDEX
C OUTPUT : (L*4)  LPLTA(,)  = .TRUE.  => PLT COEFFICIENT AVAILABLE
C                             .FALSE. => PLT COEFFICIENT NOT AVAILABLE
C                             1ST DIM: CHARGE STATE INDEX
C                             2ND DIM:  METASTABLE INDEX
C
C PROGRAM: (I*4)  IT        = GENERAL INDEX FOR TEMPERATURE
C          (I*4)  IZ        = GENERAL INDEX FOR CHARGE
C          (I*4)  IZ1       = GENERAL INDEX FOR CHARGE+1
C          (I*4)  IPRT      = GENERAL INDEX FOR PARENT METASTABLE
C          (I*4)  JPRT      = GENERAL INDEX FOR PARENT METASTABLE
C          (I*4)  IGRD      = GENERAL INDEX FOR METASTABLE
```

```
C          (I*4)  JGRD      = GENERAL INDEX FOR METASTABLE
C
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C
C
C AUTHOR : H. P. SUMMERS, JET
C          K1/1/57
C          JET  EXT. 4941
C
C DATE   : 25/04/94
C
C IDL-UNIX PORT:
C VERSION:     1.1 DATE:    31/10/95
C MODIFIED:    TIM HAMMOND
C              - INITIAL VERSION TO BE USED FOR UNIX PLATFORMS
C
C VERSION:     1.2 DATE:    08/11/95
C MODIFIED:    TIM HAMMOND
C              - ALTERED ALL DECISIONS 'IF (LEXSA(I))' WHICH OPENED ANY
C                CLASS OF FILE WHICH EXISTED EVEN IF IT HAD NOT BEEN
C                SELECTED TO 'IF (LEXSA(I).AND.LSELA(I))' SO THAT EVEN
C                IF A PARTICULAR FILE EXISTS IT IS ONLY OPENED AND READ
C                IF IT HAS BEEN REQUESTED.
C        - TIDIED UP COMMENTS AND CODE.
C
C VERSION:     1.3 DATE:    08/11/95
C MODIFIED:    TIM HAMMOND
C              - REMOVED SUPERFLUOUS VARIABLES
C
C VERSION:     1.4 DATE:    09/11/95
C MODIFIED:    TIM HAMMOND
C              - CHANGED LOOP:  2 IT = 1,ITDIMD
C                          TO:  2 IT = 1,NTDIM
C                AS IT WAS SUBSCRIPTING ARRAYS ACDA ETC. OUT OF
C                RANGE
C
C VERSION:     1.5 DATE:    13/11/95
C MODIFIED:    TIM HAMMOND
C              - CHANGED DECISION STATEMENTS AGAIN (SEE 1.2) SO THEY
C                NOW READ: 'IF ((LEXSA(I).OR.LDEFA(I)).AND.LSELA(I))'
C                AS BEFORE THE CODE WAS IGNORING THE CASE WHERE THE
C                DEFAULT FILE EXISTED AND THE USER HAD ASKED FOR THIS
C                DATA TO BE INCLUDED. IT IS NOT NECESSARY TO SPECIFY
C                DIFFERENT FILENAMES FOR DEFAULT AND USER DATA AS THE
C                ARRAY USED CONTAINS THE DEFAULT FILENAME WHEREVER
C                THE USER FILE DOES NOT EXIST. SEE d5spf0.pro FOR
C                MORE DETAILS OF THIS.
C
C-----------------------------------------------------------------------
      CHARACTER*120       DSFLLA(8)
      INTEGER             IPDIMD,      ISDIMD,      ITDIMD,      ITMAX
      INTEGER             IZ0,         IZ1MAX,      IZ1MIN,      IZDIMD
      INTEGER             NPART(IZDIMD),            NPARTR(IZDIMD)
      INTEGER             NTDIM
      LOGICAL             LACDA(IZDIMD,IPDIMD,IPDIMD)
      LOGICAL             LCCDA(IZDIMD,IPDIMD,IPDIMD)
      LOGICAL             LDEFA(8),    LEXSA(8),    LPART
      LOGICAL             LPLTA(IZDIMD,IPDIMD),     LPRBA(IZDIMD,IPDIMD)
```

```
LOGICAL            LPRCA(IZDIMD,IPDIMD)
LOGICAL            LQCDA(IZDIMD,IPDIMD,IPDIMD)
LOGICAL            LSCDA(IZDIMD,IPDIMD,IPDIMD)
LOGICAL            LSELA(8),    LXCDA(IZDIMD,IPDIMD,IPDIMD)
REAL*8             ACDA(NTDIM,IZDIMD,IPDIMD,IPDIMD)
REAL*8             CCDA(NTDIM,IZDIMD,IPDIMD,IPDIMD)
REAL*8             DDENS(ITMAX),             DDENSD(ITDIMD)
REAL*8             DRCOFD(ISDIMD,ITDIMD,ITDIMD)
REAL*8             DRCOFI(ITMAX),            DTEV(ITMAX)
REAL*8             DTEVD(ITDIMD)
REAL*8             PLTA(NTDIM,IZDIMD,IPDIMD)
REAL*8             PRBA(NTDIM,IZDIMD,IPDIMD)
REAL*8             PRCA(NTDIM,IZDIMD,IPDIMD)
REAL*8             QCDA(NTDIM,IZDIMD,IPDIMD,IPDIMD)
REAL*8             SCDA(NTDIM,IZDIMD,IPDIMD,IPDIMD)
REAL*8             XCDA(NTDIM,IZDIMD,IPDIMD,IPDIMD)
REAL*8             ZDATA(ISDIMD)
```

5.20 d5diag: Subroutine d5diag from library adas4xx

```
       SUBROUTINE D5DIAG( NDSTAT , NDMET ,
     &                    NSTATE , NMET  ,
     &                    CFREC  , CFION , CFMET
     &                  )
C
C-----------------------------------------------------------------------
C  ***************** FORTRAN77 SUBROUTINE : D5DIAG ****************      ********
C
C PURPOSE: CALCULATION OF PRIME DIAGONAL OF METASTABLE RATE COEFFICIENT
C   MATRIX
C
C CALLING PROGRAM: D5MPOP
C
C INPUT: (R*8) CFREC( , , )  = RECOMBINATION RATE COEFFICIENTS TO ALL
C                              METASTABLE NDMET;STARTING FROM FIRST TO
C                              GROUND LEVEL WITH CFREC(1,NDMET,NDMET)
C                              SET TO ZERO
C                              DIMENSIONS = (NSTATE,NDMET,NDMET)
C
C INPUT: (R*8) CFION( , , )  = IONISATION RATE COEFFICIENTS TO ALL
C                              METASTABLE NDMET;STARTING FROM GROUND
C                              TO FIRST LEVEL,WITH
C                              CFION(NSTATE,NDMET,NDMET) SET TO ZERO
C                              DIMENSIONS = (NSTATE,NDMET,NDMET)
C INPUT: (I*4) NDSTAT        = MAXIMUM NUMBER OF NDMET
C INPUT: (I*4) NSTATE        = PARAMETER = NO OF NDMET
C INPUT: (I*4) NDMET         = PARAMETER = MAXIMUM SIZE OF MATRICES
C                              HOLDING METASTABLE TRANSITIONS
C INPUT  (I*4) NMET( )       = NO  OF METASTABLES IN EACH ENERGY LEVEL
C                              DETERMINES ACTUAL SIZE OF MINI MATRICES
C                              DIMENSION = NSTATE
C
C OUTPUT:(R*8) CFMET( , , )  = CROSS COUPLING COEFFICIENTS BETWEEN
C                              METASTABLE NDMET WITH LEADING DIAGONAL
C                              CALCULATED
C                              DIMENSIONS = (NSTATE,NDMET,NDMET)
C
C ROUTINES : NONE
C
C AUTHOR : D. BROOKS,  H. P. SUMMERS, JET
C          K1/1/57
C          JET EXT. 4941
C
C
C DATE : 07/12/93
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 08-11-95
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST RELEASE
C
C-----------------------------------------------------------------------
      INTEGER             NDMET,       NDSTAT,      NMET(NDSTAT)
      INTEGER             NSTATE
      REAL*8              CFION(NDMET,NDMET,NDSTAT)
      REAL*8              CFMET(NDMET,NDMET,NDSTAT)
      REAL*8              CFREC(NDMET,NDMET,NDSTAT)
```

5.21 d5mfsp: Subroutine d5mfsp from library adas4xx

```
      SUBROUTINE D5MFSP( NDSTAT , NDMET  , NDONE  ,
     &                   NSTATE , NMET   , ID     , NMSUM  ,
     &                   CFREC  , CFION  , CFMET  , CPOPN  ,
     &                   POPN   , POPNMO , POPNPO ,
     &                   CPOPND , CPOPNZ ,
     &                   POPF   ,
     &                   XTEMP  , YTEMP  ,
     &                   RHS    , RDUM   , SOLVE  , LSOLVE , LAGAIN
     &                 )
C-----------------------------------------------------------------------------
C****************** FORTRAN77 SUBROUTINE: D5MFSP ******************** ********
C
C PURPOSE: TO PERFORM THE MAIN MATRIX ALGEBRA WHICH CALCULATES THE
C          LEVEL POPULATIONS-INCLUDING METASTABLE STATES
C
C CALLING PROGRAM: D5MPOP
C
C INPUT: (R*8) CFREC( , , )  = RECOMBINATION RATE COEFFICIENTS TO ALL
C         METASTABLE NDMET;STARTING FROM FIRST TO
C         GROUND LEVEL,WITH CFREC(NDMET,NDMET,1)
C         SET TO ZERO
C         DIMENSIONS = (NDMET,NDMET,NDSTAT)
C
C        (R*8) CFION( , , )  = IONISATION RATE COEFFICIENTS TO ALL
C         METASTABLE NDMET;STARTING FROM GROUND TO
C         FIRST LEVEL,WITH
C                CFION(NDMET,NDMET,NSTATE) SET TO ZERO
C         DIMENSIONS = (NDMET,NDMET,NDSTAT)
C
C        (I*4) NSTATE        = PARAMETER = NO OF NDMET
C
C        (I*4) NDMET         = PARAMETER = MAXIMUM SIZE OF MATRICES
C         HOLDING METASTABLE TRANSITIONS
C
C        (R*8) NMET( )       = NO  OF METASTABLES IN EACH ENERGY LEVEL
C         DETERMINES ACTUAL SIZE OF MINI MATRICES
C         DIMENSION = NDSTAT
C
C        (I*4) NDONE         = 1 0;MODIFYING MATRICES IN ORDER TO USE
C         SUBROUTINES
C
C        (I*4) ID            = POSITION OF DOMINANT TERM
C
C        (R*8) CFMET( , , )  = CROSS COUPLING COEFFICIENTS BETWEEN
C         METASTABLE NDMET WITH LEADING DIAGONAL
C         CALCULATED
C         DIMENSIONS = (NDMET,NDMET,NDSTAT)
C
C        (L*4) LSOLVE        =  TRUE  => XXMINV SOLVES SET OF 'N' LINEAR
C          EQUATIONS A X = B WHERE A,X,B ARE
C          MATRICES/VECTORS AND:
C          A = 'A(,)' ON INPUT
C          X = 'B()' ON OUTPUT
C          B = 'B()' ON INPUT
C          FALSE  => ONLY MATRIX INVERSION,
C          A INVERSE REPLACES A
C
C        (R*8) DINT          = + OR - 1 DEPENDING ON THE NUMBER OF ROW
C         INTERCHANGES IN THE MATRIX INVERSION
```

```
C
C
C        (R*8) NMETZ    = ACTUAL DIMENSION OF NORMALIZATION MATRIX
C         ONCE FIRST ROW & COLUMN IS ELIMINATED
C       = NMET(ID)+NMET(ID+1)-1
C
C        (I*4) NDSTAT    = PARAMETER = MAXIMUM NUMBER OF NDMET
C
C        (I*4) NPOSX    = NMET(ID)
C
C        (I*4) NPOSY    = NMET(ID+1)
C
C
C OUTPUT:(R*8) CPOPN( , , )  = ARRAY HOLDING COEFFICIENTS OF POPULATION
C         STATE EQUATIONS
C         DIMENSIONS = (NDMET,NDMET,NDSTAT+1)
C
C        (R*8) POPN( , , )   = ARRAY HOLDING POPULATION STATE VALUES WITH
C         THIRD DIMENSION SET TO 1
C         DIMENSIONS = (NDMET,NDONE,NDSTAT+1)
C
C        (R*8) CPOPND( , , ) = TEMPORARY NAME OF MATRIX TO BE SUBSTITUTED
C         INTO NEXT EQUATION IN UPWARD LOOP
C         DIMENSIONS = (NDMET,NDMET,NDSTAT+1)
C
C        (R*8) CPOPNZ( , , ) = TEMPORARY NAME OF MATRIX TO BE SUBSTITUTED
C         INTO NEXT EQUATION  IN DOWNWARD LOOP
C         DIMENSIONS = (NDMET,NDMET,NDSTAT+1)
C
C        (R*8) POPNPO( , , ) = TEMPORARY NAME OF MATRIX HOLDING POPULATION
C         STATE VALUES AFTER NORMALIZATION,TO BE
C         SUBSTITUTED INTO NEXT EQUATION IN UPWARD
C         LOOP
C         DIMENSIONS = (NDMET,NDONE,NDSTAT+1)
C
C        (R*8) POPNMO( , , ) = TEMPORARY NAME OF MATRIX HOLDING POPULATION
C         STATE VALUES AFTER NORMALIZATION,TO BE
C         SUBSTITUTED INTO NEXT EQUATION IN DOWNWARD
C         LOOP
C         DIMENSIONS = (NDMET,NDONE,NDSTAT+1)
C
C        (R*8) SUM           = SUM OF ALL LEVEL POPULATION VALUES
C         INCLUDING METASTABLES
C
C        (R*8) XTEMP( , )    = TEMPORARY MATRIX FOR DURING SUBROUTINE
C         CALCULATIONS
C         DIMENSIONS = (NDMET,NDMET)
C
C        (R*8) YTEMP( , )    = TEMPORARY MATRIX FOR DURING SUBROUTINE
C         CALCULATIONS
C         DIMENSIONS = (NDMET,NDMET)
C
C        (R*8) PTEMP( , , )  = TEMPORARY MATRIX FOR DURING ERROR CHECK
C         CALCULATIONS
C         DIMENSIONS = (NDMET,NDONE,NDSTAT)
C
C        (R*8) QTEMP( , , )  = TEMPORARY MATRIX FOR DURING ERROR CHECK
C         CALCULATIONS
C         DIMENSIONS = (NDMET,NDONE,NDSTAT)
C
C        (R*8) RTEMP( , , )  = TEMPORARY MATRIX FOR DURING ERROR CHECK
```

```
C         CALCULATIONS
C         DIMENSIONS = (NDMET,NDONE,NDSTAT)
C
C        (R*8) STEMP( , , )  = TEMPORARY MATRIX FOR DURING ERROR CHECK
C         CALCULATIONS
C         DIMENSIONS = (NDMET,NDONE,NDSTAT)
C
C        (R*8) TEMP( , , )   = MATRIX HOLDING RESULTS OF ERROR CHECK
C         ALL OF WHICH SHOULD BE ZERO
C         DIMENSIONS = (NDMET,NDONE,NDSTAT)
C
C        (R*8) SOLVE( , )    = NORMALIZATION MATRIX AT CRITICAL STAGE
C         DIMENSIONS = (2*NDMET-1,2*NDMET-1)
C
C        (R*8) CTEMP( , , )  = HOLDS VALUES OF CFMET FOR ERROR CHECK,IS
C         NECCESSARY SINCE CFMET IS ALTERED DURING
C         CALCULATIONS
C         DIMENSIONS = (NDMET,NDMET,NDSTAT)
C
C        (R*8) RHS( )    = SIPHONED OFF COLUMN OF NORMALIZATION
C         MATRIX,USED TO CALCULATE METASTABLE
C         NDMET OF DOMINANT STAGE THROUGH XXMINV
C         DIMENSIONS = (2*NDMET-1)
C
C        (R*8) RDUM( )    = DUMMY ARRAY USED IN XXMINV AS RHS WHEN
C         LSOLVE =  FALSE
C
C
C ROUTINES :
C           ROUTINE   SOURCE    BRIEF DESCRIPTION
C           ____________________________________________________________
C           DXMADD    ADAS      MATRIX ADDITION/SUBTRACTION
C           DXMMUL    ADAS      MATRIX MULTIPLICATION
C           XXMINV    ADAS      MATRIX INVERSION
C
C
C AUTHOR:  D. BROOKS, H. P. SUMMERS, JET
C          K1/1/57
C          JET EXT. 4941
C
C DATE:    02/06/94
C
C UPDATE:  14/02/95  HPS - INTRODUCED IAGAIN TO IMPROVE DOMINANT STAGE
C                          IDENTIFICATION.
C UPDATE:  06/07/95  HPS - MODIFIED LOOP TO PREVENT IAGAIN AND HENCE ID
C                          BEING SET GREATER THAN NSTATE-1.
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 08-11-95
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST RELEASE
C
C VERSION: 1.2                          DATE: 01-12-95
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - COPIED FOLLOWING UPDATES MADE BY DAVID BROOKS:
C
C UPDATE:  29/11/95  DHB - INTRODUCED A CHECK TO MAKE SURE THAT THE
C                          POPULATION EQUATIONS ARE SOLVED FOR THE BEST
C                          POSSIBLE CHOICE OF DOMINANT STAGE. NB: THIS
C                          IS NOT THE DOMINANT STAGE ITSELF BUT THE
```

```
C                          NEAREST STAGE TO IT THAT CAN SUPPORT THE
C                          CALCULATION I.E. THERE IS A CHECK TO MAKE
C                          SURE THE DOMINANT STAGE IDENTIFICATION DOES
C                          NOT PUSH THE SOLUTION LOOP TOO CLOSE TO ANY
C                          REGION OF RAPID POPULATION DROP OFF. THE
C                          PARAMETER ACC HAS BEEN INTRODUCED TO MEASURE
C                          THIS DROP OFF AND CAN BE ADJUSTED IF IT IS TOO
C                          STRINGENT.
C UPDATE:  29/11/95  DHB - ADDED CHECK TO AVOID UNNECCESARY LOOPING IN
C                          UNRESOLVED CASE.
C
C VERSION: 1.3                          DATE: 01-12-95
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - SWAPPED ORDER OF DECLARATION OF PARAMETER ACC.
C
C-----------------------------------------------------------------------------
      INTEGER             ID,          NDMET,       NDONE,       NDSTAT
      INTEGER             NMET(NDSTAT),             NMSUM,       NSTATE
      LOGICAL             LAGAIN,      LSOLVE
      REAL*8              CFION(NDMET,NDMET,NDSTAT)
      REAL*8              CFMET(NDMET,NDMET,NDSTAT)
      REAL*8              CFREC(NDMET,NDMET,NDSTAT)
      REAL*8              CPOPN(NDMET,NDMET,NDSTAT+1)
      REAL*8              CPOPND(NDMET,NDMET,NDSTAT+1)
      REAL*8              CPOPNZ(NDMET,NDMET,NDSTAT+1)
      REAL*8              POPF(NMSUM), POPN(NDMET,NDONE,NDSTAT+1)
      REAL*8              POPNMO(NDMET,NDONE,NDSTAT+1)
      REAL*8              POPNPO(NDMET,NDONE,NDSTAT+1)
      REAL*8              RDUM(NDMET), RHS(2*NDMET-1)
      REAL*8              SOLVE(2*NDMET-1,2*NDMET-1)
      REAL*8              XTEMP(NDMET,NDMET),       YTEMP(NDMET,NDMET)
```

5.22 d5mpop: Subroutine d5mpop from library adas4xx

```
       SUBROUTINE D5MPOP( NTDIM , IZDIMD, IPDIMD,
     &                    NSTAGE, ITMAX , NPRT  , NMSUM ,
     &                    ACDA  , SCDA  , CCDA  , QCDA  , XCDA  ,
     &                    DENS  , DENSH ,
     &                    ITEM  ,
     &                    CFREC , CFION , CFMET ,
     &                    POPN  , POPNMO, POPNPO,
     &                    CPOPN , CPOPND, CPOPNZ,
     &                    POPF  ,
     &                    XTEMP , YTEMP , YTEM  ,
     &                    RHS   , RDUM  , SOLVE , LSOLVE
     &                   )
       IMPLICIT REAL*8(A-H,O-Z)
C
C-----------------------------------------------------------------------------
C                                                                       ********
C  ****************** FORTRAN 77 SUBROUTINE: D5MPOP ***************************
C
C  PURPOSE: CALCULATION OF METASTABLE RESOLVED IONISATION STAGE
C           POPULATIONS OF A PARTICULAR ELEMENT FOR A GIVEN TEMPERATURE
C           AND DENSITY
C
C  CALLING PROGRAM: ADAS405
C
C  SUBROUTINE:
C
C
C INPUT :(I*4) NTDIM         = MAXIMUM NUMBER OF MODEL TEMPS/DENSITIES
C INPUT :(I*4) IZDIMD        = MAXIMUM NUMBER OF STAGES-1
C INPUT :(I*4) IPDIMD        = MAXIMUM SIZE OF METASTABLES FRO A STAGE
C INPUT :(I*4) NSTAGE        = NUMBER OF STAGES-1
C INPUT :(I*4) ITMAX         = NUMBER OF MODEL TEMPS/DENSITIES
C INPUT :(R*8) NPRT( )       = PARTITION OF TOTAL METASTABLES ACCORDING
C                              TO IONISATION STAGES
C                              1ST DIM: STAGE INDEX
C INPUT :(I*4) NMSUM         = TOTAL NUMBER OF POPULATIONS
C INPUT :(R*8) ACDA(,,,)     = GENERALISED CR RECOMBINATION COEFFICIENT
C                              1ST DIM: TEMPERATURE INDEX
C                              2ND DIM: STAGE INDEX (LESS 1)
C                              3RD DIM: METASTABLE INDEX
C                              4TH DIM: METASTABLE INDEX
C INPUT :(R*8) SCDA(,,,)     = GENERALISED CR IONISATION COEFFICIENT
C                              1ST DIM: TEMPERATURE INDEX
C                              2ND DIM: STAGE INDEX (LESS 1)
C                              3RD DIM: METASTABLE INDEX
C                              4TH DIM: METASTABLE INDEX
C INPUT :(R*8) CCDA(,,,)     = GENERALISED CR CHARGE EXCH. COEFFICIENT
C                              1ST DIM: TEMPERATURE INDEX
C                              2ND DIM: STAGE INDEX (LESS 1)
C                              3RD DIM: METASTABLE INDEX
C                              4TH DIM: METASTABLE INDEX
C INPUT :(R*8) QCDA(,,,)     = GENERALISED CR CROSS-COUPL. COEFFICIENT
C                              1ST DIM: TEMPERATURE INDEX
C                              2ND DIM: STAGE INDEX (LESS 1)
C                              3RD DIM: METASTABLE INDEX
C                              4TH DIM: METASTABLE INDEX
C INPUT :(R*8) XCDA(,,,)     = GENERALISED CR PARENT X-CP. COEFFICIENT
C                              1ST DIM: TEMPERATURE INDEX
C                              2ND DIM: STAGE INDEX (LESS 1)
```

```
C                               3RD DIM: METASTABLE INDEX
C                               4TH DIM: METASTABLE INDEX
C INPUT :(R*8) DENS()         = ELECTRON DENSITIES FOR MODEL
C INPUT :(R*8) DENSH()        = NEUTRAL HYDROGEN DENSITIES FOR MODEL
C
C INPUT :(I*4) ITEM           = CURRENT TEMP/DENSITY INDEX
C
C OUTPUT:(R*8) CFREC( , , )   = RECOMBINATION RATE COEFFICIENTS TO ALL
C                               METASTABLE IPDIMD;STARTING FROM FIRST TO
C                               GROUND LEVEL,WITH CFREC(1,IPDIMD,IPDIMD)
C                               SET TO ZERO
C                               DIMENSIONS = (IPDIMD,IPDIMD,IZDIMD)
C OUTPUT:(R*8) CFION( , , )   = IONISATION RATE COEFFICIENTS TO ALL
C                               METASTABLE IPDIMD;STARTING FROM GROUND
C                               TO FIRST LEVEL,WITH
C                               CFION(NSTAGE,IPDIMD,IPDIMD)
C                               SET TO ZERO
C                               DIMENSIONS = (IPDIMD,IPDIMD,IZDIMD)
C OUTPUT:(R*8) CFMET( , , )   = CROSS COUPLING COEFFICIENTS BETWEEN
C                               METASTABLE IPDIMD WITH LEADING DIAGONAL
C                               CALCULATED
C                               DIMENSIONS = (IPDIMD,IPDIMD,IZDIMD)
C
C OUTPUT:(R*8) POPN( , , )    = ARRAY HOLDING POPULATION STATE VALUES
C                               WITH SECOND DIMENSION SET TO 1
C                               DIMENSIONS = (IPDIMD,NDONE,IZDIMD+1)
C OUTPUT:(R*8) POPNMO( , , ) =TEMPORARY NAME OF MATRIX HOLDING POPULATI ON
C                               STATE VALUES AFTER NORMALIZATION,TO BE
C                               SUBSTITUTED INTO NEXT EQUATION IN
C                               DOWNWARD LOOP
C                               DIMENSIONS = (IPDIMD,NDONE,IZDIMD+1)
C OUTPUT:(R*8) POPNPO( , , ) =TEMPORARY NAME OF MATRIX HOLDING POPULATI ON
C                               STATE VALUES AFTER NORMALIZATION,TO BE
C                               SUBSTITUTED INTO NEXT EQUATION IN UPWARD
C                               LOOP
C                               DIMENSIONS = (IPDIMD,NDONE,IZDIMD+1)
C
C OUTPUT:(R*8) CPOPN( , , )   = ARRAY HOLDING COEFFICIENTS OF POPULATION
C                               STATE EQUATIONS
C                               DIMENSIONS = (IPDIMD,IPDIMD,IZDIMD+1)
C OUTPUT:(R*8) CPOPND( , , ) = TEMPORARY NAME OF MATRIX TO BE SUBSTITUTED
C                               INTO NEXT EQUATION IN UPWARD LOOP
C                               DIMENSIONS = (IPDIMD,IPDIMD,IZDIMD+1)
C OUTPUT:(R*8) CPOPNZ( , , ) = TEMPORARY NAME OF MATRIX TO BE SUBSTITUTED
C                               INTO NEXT EQUATION  IN DOWNWARD LOOP
C                               DIMENSIONS = (IPDIMD,IPDIMD,IZDIMD+1)
C
C OUTPUT:(R*8) POPF()         = POPULATIONS FOR A SPECIFIED TEMPERATURE  D
C                               1ST DIM: INDEX OVER STAGES/METASTABLES
C
C OUTPUT:(R*8) XTEMP( , )     =TEMPORARY MATRIX USED DURING SUBROUTINE
C                               CALCULATIONS
C                               DIMENSIONS = (IPDIMD,IPDIMD)
C OUTPUT:(R*8) YTEMP( , )     =TEMPORARY MATRIX FOR DURING SUBROUTINE
C                               CALCULATIONS
C                               DIMENSIONS = (IPDIMD,IPDIMD)
C OUTPUT:(R*8) YTEM( )        = TEMPORARY ARRAY FOR HOLDING VALUES OF
C                               DIFFERENCE BETWEEN RECOMBINATION AND
C                               IONISATION GROUND LEVEL COEFFICIENTS
C                               DIMENSIONS = (NSTAGE)
C
```

```
C OUTPUT:(R*8) RHS( )        = SIPHONED OFF COLUMN OF NORMALIZATION
C                              MATRIX,USED TO CALCULATE METASTABLE
C                              IPDIMD OF DOMINANT STAGE THROUGH MATINV
C                              DIMENSIONS = (2*IPDIMD-1)
C OUTPUT:(R*8) RDUM( )       = DUMMY ARRAY USED IN MATINV AS RHS WHEN
C                              LSOLVE =  FALSE
C OUTPUT:(R*8) SOLVE( , )    = NORMALIZATION MATRIX AT CRITICAL STAGE
C                              DIMENSIONS = (2*IPDIMD-1,2*IPDIMD-1)
C OUTPUT:(L*4) LSOLVE        = .TRUE.  => SOLVE SET OF EQUATIONS
C                            = .FALSE. => INVERT MATRIX ONLY
C
C        (I*4) NDONE         = PARAMETER = 1 TO ALLOW 3D MATRIX USE
C        (I*4) ID            = POSITION OF DOMINANT TERM
C        (I*4) ISTATE        = STAGE INDEX
C        (I*4) ITEM          = GENERAL INDEX
C        (I*4) I             = GENERAL INDEX
C        (I*4) J             = GENERAL INDEX
C        (I*4) K             = GENERAL INDEX
C        (R*8) YMIN          = VALUE OF DIFFERENCE BETWEEN
C                               RECOMBINATION AND IONISATION COEFFICIENTS
C                               OF GROUND IPDIMD
C
C  ROUTINES:
C           ROUTINE    SOURCE    BRIEF DESCRIPTION
C           ------------------------------------------------------
C           D5DIAG     ADAS      SETS UP ON-DIAGONAL ELEMENT OF MATRIX
C           D5MFSP     ADAS      EXECUTES PARTITION MATRIX INVERSION
C           DXMADD     ADAS      MATRIX ADDITION/SUBTRACTION
C           DXMMUL     ADAS      MATRIX MULTIPLICATION
C           XXMINV     ADAS      MATRIX INVERSION
C
C
C  AUTHOR:  D. BROOKS, H. P. SUMMERS, JET
C           K1/1/57
C           JET EXT. 4941
C
C  DATE:    02/06/94
C
C  UPDATE:  14/02/95  HPS - INTRODUCED IAGAIN TO IMPROVE DOMINANT STAGE
C                           IDENTIFICATION.  CHANGED A LOOP LIMIT.
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 08-11-95
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST RELEASE
C
C VERSION: 1.2                          DATE: 01-12-95
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - COPIED FOLLOWING CHANGES MADE BY DAVID BROOKS
C
C  UPDATE:  29/11/95  DHB - CHANGED THE METHOD FOR THE FIRST GUESS AT THE
C                           DOMINANT STAGE TO AVOID DIVISION BY ZERO
C                           ERRORS IN D5MFSP. NOW PICK A STAGE THAT HAS AN
C                           INVERTIBLE SOLUTION AND ADJUST AFTER THE
C                           POPULATION FRACTIONS HAVE BEEN CALCULATED.
C
C-----------------------------------------------------------------------
      INTEGER             IPDIMD,      ITEM,        ITMAX,       IZDIMD
      INTEGER             NMSUM,       NPRT(IZDIMD),             NSTAGE
      INTEGER             NTDIM
      LOGICAL             LSOLVE
```

```
REAL*8            ACDA(NTDIM,IZDIMD,IPDIMD,IPDIMD)
REAL*8            CCDA(NTDIM,IZDIMD,IPDIMD,IPDIMD)
REAL*8            CFION(IPDIMD,IPDIMD,IZDIMD)
REAL*8            CFMET(IPDIMD,IPDIMD,IZDIMD)
REAL*8            CFREC(IPDIMD,IPDIMD,IZDIMD)
REAL*8            CPOPN(IPDIMD,IPDIMD,IZDIMD+1)
REAL*8            CPOPND(IPDIMD,IPDIMD,IZDIMD+1)
REAL*8            CPOPNZ(IPDIMD,IPDIMD,IZDIMD+1)
REAL*8            DENS(NTDIM), DENSH(NTDIM)
REAL*8            POPF(NMSUM), POPN(IPDIMD,NDONE,IZDIMD+1)
REAL*8            POPNMO(IPDIMD,NDONE,IZDIMD+1)
REAL*8            POPNPO(IPDIMD,NDONE,IZDIMD+1)
REAL*8            QCDA(NTDIM,IZDIMD,IPDIMD,IPDIMD)
REAL*8            RDUM(IPDIMD),             RHS(2*IPDIMD-1)
REAL*8            SCDA(NTDIM,IZDIMD,IPDIMD,IPDIMD)
REAL*8            SOLVE(2*IPDIMD-1,2*IPDIMD-1)
REAL*8            XCDA(NTDIM,IZDIMD,IPDIMD,IPDIMD)
REAL*8            XTEMP(IPDIMD,IPDIMD),     YTEM(IZDIMD)
REAL*8            YTEMP(IPDIMD,IPDIMD)
```

5.23 d5scrp: Subroutine d5scrp from library adas4xx

```
       SUBROUTINE D5SCRP( LRSCRP , LSNULL  ,
     &                    DSNINC , DSPECA  ,
     &                    NDLINE , NDCOMP  , NDRAT  , NDFILE ,
     &                    NFILE  , LFILE   ,
     &                    UID    , GROUP   , TYPE   , EXT    , ION    ,
     &                    MEMB   , IZ0     ,
     &                    NLINE  , NCOMP   ,
     &                    IZION  , IMET    , CIMET  , INDPH  , CINDPH ,
     &                    IFILE  , TITL    ,
     &                    NRAT   ,
     &                    ILINE  , JLINE   , TITR   , IRCODE
     &                  )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: D5SCRP  *********************
C
C  PURPOSE: TO READ SCRIPT FILE AND ACCESS EMISSIVITY DATA
C           ON SPECTRAL LINES REQUESTED FOR FURTHER PROCESSING IN
C           EQUILIBRIUM IONISATION CODES.
C
C  CALLING PROGRAM: ADAS405
C
C  SUBROUTINE:
C
C  INPUT : (C*120) DSNINC   = SCRIPT DATA SET NAME (FULL MVS DSN)
C                             (IN FORM SUITABLE FOR DYNAMIC ALLOCATION)
C  INPUT : (I*4)   NDLINE   = MAXIMUM NUMBER OF LINES ALLOWED
C  INPUT : (I*4)   NDCOMP   = MAXIMUM NUMBER OF COMPONENT FOR EACH LINE
C  INPUT : (I*4)   NDRAT    = MAXIMUM NUMBER OF LINE RATIOS ALLOWED
C  INPUT : (I*4)   NDFILE   = MAXIMUM NUMBER OF EMISSIVITY FILES WHICH
C                             CAN BE SEARCHED
C
C  OUTPUT: (L*4)   LRSCRP   = .TRUE.  => SCRIPT FILE READ
C                             .FALSE. => SCRIPT FILE NOT READ
C  OUTPUT: (L*4)   LSNULL   = .TRUE.  => SCRIPT FILE SET TO NULL
C                             .FALSE. => SCRIPT FILE VALID
C  OUTPUT: (C*120) DSPECA() = PHOTON EMISSIVITY SOURCE FILES
C  OUTPUT: (I*4)   NFILE    = NUMBER OF PEC FILES TO BE SCANNED
C  OUTPUT: (L*4)   LFILE()  = .TRUE.  => PEC FILE EXISTS AND MATCHES
C                             .FALSE. => PEC FILE DOES NOT EXIST/MATCH
C  OUTPUT: (C*6)   UID()    = USER IDENTIFIER OF PEC FILE
C  OUTPUT: (C*8)   GROUP()  = GROUP IDENTIFIER OF PEC FILE
C  OUTPUT: (C*5)   TYPE()   = TYPE IDENTIFIER OF PEC FILE
C  OUTPUT: (C*3)   EXT()    = EXTENSION OF PEC FILE MEMBER NAME
C  OUTPUT: (C*4)   ION()    = ION NAME OF PEC FILE MEMBER NAME
C  OUTPUT: (C*8)   MEMB()   = MEMBER NAME OF PEC FILE
C  OUTPUT: (I*4)   NLINE    = NUMBER OF LINES IDENTIFIED IN SCRIPT
C  OUTPUT: (I*4)   NCOMP()  = NUMBER OF COMPONENTS OF SCRIPT LINE
C                             1ST DIM: LINE INDEX
C  OUTPUT: (I*4)   IZION(,) = CHARGE STATE OF COMPONENT
C                             1ST DIM: LINE INDEX
C                             2ND DIM: COMPONENT INDEX
C  OUTPUT: (I*4)   IMET(,)  = NUMBER OF COMPONENTS OF SCRIPT LINE
C                             1ST DIM: LINE INDEX
C                             2ND DIM: COMPONENT INDEX
C  OUTPUT: (C*1)   CIMET(,) = SIGN (+, BLANK OR -) OF METASTABLE
C                             1ST DIM: LINE INDEX
C                             2ND DIM: COMPONENT INDEX
C  OUTPUT: (I*4)   INDPH(,) = PEC FILE INDEX OF LINE COMPONENT
```

```
C                              1ST DIM: LINE INDEX
C                              2ND DIM: COMPONENT INDEX
C  OUTPUT: (C*1)   CINDPH(,)= DRIVER (E OR BLANK => ELECTRONS)
C                                    (H          => HYDROGEN )
C                              1ST DIM: LINE INDEX
C                              2ND DIM: COMPONENT INDEX
C  OUTPUT: (I*4)   IFILE(,) = INDEX OF PEC FILE IN FILE LIST
C                              1ST DIM: LINE INDEX
C                              2ND DIM: COMPONENT INDEX
C  OUTPUT: (C*12)  TITL(,)  = TITLE FOR LINE COMPONENT
C                              1ST DIM: LINE INDEX
C                              2ND DIM: COMPONENT INDEX
C  OUTPUT: (I*4)   NRAT     = NUMBER OF RATIOS IDENTIFIED IN SCRIPT
C  OUTPUT: (I*4)   ILINE()  = INDEX OF NUMERATOR LINE FOR LINE RATIO
C  OUTPUT: (I*4)   JLINE()  = INDEX OF DENOMINATOR LINE FOR LINE RATIO
C  OUTPUT: (C*25)  TITR()   = TILE FOR LINE RATIO
C  OUTPUT: (I*4)   IRCODE   = ERROR FLAG:
C                              0 => SCRIPT FILE WAS READ OKAY
C                              1 => SCRIPT FILE DOES NOT EXIST
C                              2 => I/O ERROR READING THE SCRIPT FILE
C                              3 => 1 OR MORE FILE NAMES IN SCRIPT FILE
C                                   IS/ARE INVALID.
C
C          (I*4)   IUNT10  = PARAMETER = INPUT UNIT FOR DATA
C          (L*4)   OPEN10  = .TRUE.  => FILE ALLOCATED TO UNIT 10.
C                            .FALSE. => NO FILE ALLOCATED TO UNIT 10.
C
C ROUTINES:
C          ROUTINE    SOURCE   BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          XXSLEN     ADAS      FIND NON-BLANK CHARACTERS IN STRING
C
C
C AUTHOR:  H. P. SUMMERS, JET
C          K1/1/57
C          JET EXT. 4941
C
C DATE:    20/04/94
C
C IDL-UNIX PORT:
C VERSION: 1.1                  DATE:    08/11/95
C MODIFIED: TIM HAMMOND
C               - ADDED SCCS HEADER - FIRST VERSION
C
C VERSION: 1.2                  DATE:    08/11/95
C MODIFIED: TIM HAMMOND
C               - REMOVED SUPERFLUOUS VARIABLES
C
C VERSION: 1.3                  DATE:    10/11/95
C MODIFIED: TIM HAMMOND
C               - ADDED NEW ERROR CHECK VARIABLE IRCODE TO SEE
C                 WHETHER THE SCRIPT FILE WAS READ OKAY OR NOT.
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      CHARACTER           CIMET(NDLINE,NDCOMP)
      CHARACTER           CINDPH(NDLINE,NDCOMP)
      CHARACTER*120       DSNINC,      DSPECA(NDFILE)
      CHARACTER*3         EXT(NDFILE)
      CHARACTER*8         GROUP(NDFILE)
```

```
CHARACTER*4         ION(NDFILE)
CHARACTER*8         MEMB(NDFILE)
CHARACTER*12        TITL(NDLINE,NDCOMP)
CHARACTER*25        TITR(NDRAT)
CHARACTER*5         TYPE(NDFILE)
CHARACTER*6         UID(NDFILE)
INTEGER             IFILE(NDLINE,NDCOMP),     ILINE(NDRAT)
INTEGER             IMET(NDLINE,NDCOMP),      INDPH(NDLINE,NDCOMP)
INTEGER             IRCODE,      IZ0,         IZION(NDLINE,NDCOMP)
INTEGER             JLINE(NDRAT),             NCOMP(NDLINE)
INTEGER             NDCOMP,      NDFILE,      NDLINE,      NDRAT
INTEGER             NFILE,       NLINE,       NRAT
LOGICAL             LFILE(NDFILE),            LRSCRP,      LSNULL
```

5.24 d5sgcf: Subroutine d5sgcf from library adas4xx

```
       SUBROUTINE D5SGCF( IZ0    , IZL    , IZH    ,
     &                    ISDIMD , IZDIMD , ITDIMD , IPDIMD , IMDIMD ,
     &                    NMSUM  , IZIP   , IMIP   , IPIZM  ,
     &                    NDLINE , NDCOMP ,
     &                    NLINE  , NCOMP  , SPECL  , IPLINE ,
     &                    IZION  , IMET   , CIMET  , INDPH  , CINDPH ,
     &                    IFILE  ,
     &                    NTDIM  , ITMAX  ,
     &                    DENS   , DENSH  ,
     &                    PECA   , LPEC   ,
     &                    FPABUN ,
     &                    GCFPEQ , GCFEQ  ,
     &                    NDRAT  , NRAT   ,
     &                    ILINE  , JLINE  ,
     &                    RATA
     &                  )
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: D5SGCF *********************
C
C PURPOSE : TO ASSEMBLE GCF FUNCTIONS AND THEIR COMPONENTS USING
C           FRACTIONAL METASTABLE ABUNDANCES.
C
C
C
C INPUT  : (I*4)  IZ0       = NUCLEAR CHARGE
C INPUT  : (I*4)  IZL       = MINIMUM ION CHARGE+1 IN MASTER DATA FILES
C INPUT  : (I*4)  IZH       = MAXIMUM ION CHARGE+1 IN MASTER DATA FILES
C INPUT  : (I*4)  ISDIMD    = MAXIMUM NUMBER OF (CHARGE, PARENT, GROUND)
C                             BLOCKS IN ISONUCLEAR MASTER FILES
C INPUT  : (I*4)  IZDIMD    = MAXIMUM NUMBER OF CHARGE STATES
C                             IN ISONUCLEAR MASTER FILES
C INPUT  : (I*4)  ITDIMD    = MAXIMUM NUMBER OF TEMP OR DENS VALUES IN
C                             ISOELECTRONIC MASTER FILES
C INPUT  : (I*4)  IPDIMD    = MAXIMUM NUMBER OF METASTABLES FOR EACH
C                             IONISATION STAGE
C INPUT  : (I*4)  IMDIMD    = MAXIMUM NUMBER OF METASTABLES
C
C INPUT  : (I*4)  NMSUM     = TOTAL NUMBER OF POPULATIONS
C
C INPUT  :        IZIP()    = ION CHARGE +1 (IZ1) OF METASTABLE IN LIST
C INPUT  :        IMIP()    = METASTABLE INDEX WITHIN CHARGE STATE IZ1
C                             OF METASTABLE INDEX FROM COMPLETE LIST
C INPUT  :        IPIZM(,)  = METASTABLE INDEX IN COMPLETE LIST
C                             1ST DIM: INDEX IZ1-IZL+1
C                             2ND DIM: METASTABLE COUNT FOR STAGE (IGRD)
C INPUT  : (I*4)  NDLINE    = MAXIMUM NUMBER OF LINES ALLOWED
C INPUT  : (I*4)  NDCOMP    = MAXIMUM NUMBER OF COMPONENT FOR EACH LINE
C INPUT  : (I*4)  NLINE     = NUMBER OF LINES IDENTIFIED IN SCRIPT
C INPUT  : (I*4)  NCOMP()   = NUMBER OF COMPONENTS OF SCRIPT LINE
C INPUT  : (I*4)  IZION(,)  = CHARGE STATE OF COMPONENT
C                             1ST DIM: LINE INDEX
C                             2ND DIM: COMPONENT INDEX
C INPUT  : (I*4)  IMET(,)   = METASTABLE INDEX OF COMPONENT OF
C                             SCRIPT LINE WITHIN CHARGE STATE
C                             1ST DIM: LINE INDEX
C                             2ND DIM: COMPONENT INDEX
C INPUT  : (C*1)  CIMET(,)  = SIGN (+, BLANK OR -) OF METASTABLE
```

```
C                               1ST DIM: LINE INDEX
C                               2ND DIM: COMPONENT INDEX
C INPUT  : (I*4)  INDPH(,)   = PEC FILE INDEX OF LINE COMPONENT
C                               1ST DIM: LINE INDEX
C                               2ND DIM: COMPONENT INDEX
C INPUT  : (C*1   CINDPH(,) = DRIVER (E OR BLANK => ELECTRONS)
C                                     (H          => HYDROGEN )
C                                1ST DIM: LINE INDEX
C                                2ND DIM: COMPONENT INDEX
C INPUT  : (I*4)  IFILE(,)   = INDEX OF PEC FILE IN FILE LIST
C                               1ST DIM: LINE INDEX
C                               2ND DIM: COMPONENT INDEX
C INPUT  : (I*4)  NTDIM      = MAXIMUM NUMBER OF DTEV/DDENS PAIRS
C INPUT  : (I*4)  ITMAX      = NUMBER OF ( DTEV() , DDENS() ) PAIRS
C INPUT  : (R*8)  DENS()     = ELECTRON DENSITIES (CM-3))
C INPUT  : (R*8)  DENSH()    = HYDROGEN DENSITIES (CM-3))
C INPUT  : (R*8)  PECA(,,)   = PHOTON EMISSIVITY COEFFICIENTS (CM3 S-1)
C                               1ST DIM: TEMPERATURE INDEX
C                               2ND DIM: LINE INDEX
C                               3RD DIM: COMPONENT INDEX
C INPUT  : (L*4)  LPEC(,)    = .TRUE.  => PHOTON EMISSIVITY OBTAINED
C                               .FALSE. => PHOTON EMISSIVITY NOT OBTAINED
C                               2ND DIM: LINE INDEX
C                               3RD DIM: COMPONENT INDEX
C INPUT  : (R*8)  FPABUN(,) = RESOLVED METASTABLE EQUILIBRIUM
C                               FRACTIONAL ABUNDANCES
C                               1ST DIM: - TEMPERATURE/DENSITY PAIR
C                               2ND DIM: - METASTABLE INDEX
C INPUT  : (I*4)  NDRAT      = MAXIMUM NUMBER OF LINE RATIOS ALLOWED
C INPUT  : (I*4)  NRAT       = NUMBER OF RATIOS IDENTIFIED IN SCRIPT
C INPUT  : (I*4)  ILINE()    = INDEX OF NUMERATOR LINE FOR LINE RATIO
C INPUT  : (I*4)  JLINE()    = INDEX OF DENOMINATOR LINE FOR LINE RATIO
C
C OUTPUT : (C*16) SPECL(,)   = SPEC. OF POINTERS OF LINE COMPONENT
C                               1ST DIM: LINE INDEX
C                               2ND DIM: COMPONENT INDEX
C OUTPUT : (I*4)  IPLINE(,) = METASTABLE POINTER OF LINE COMPONENT
C                               1ST DIM: LINE INDEX
C                               2ND DIM: COMPONENT INDEX
C OUTPUT : (R*8)  GCFPEQ(,,)= GCF FUNC. COMPONENT (CM3 S-1)
C                               1ST DIM: TEMPERATURE INDEX
C                               2ND DIM: LINE INDEX
C                               3ND DIM: LINE COMPONENT INDEX
C OUTPUT : (R*8)  GCFEQ(,)   = GCF FUNCTION  (CM3 S-1)
C                               1ST DIM: TEMPERATURE INDEX
C                               2ND DIM: LINE INDEX
C OUTPUT : (R*8)  RATA(,)    = LINE GCF RATIOS
C                               1ST IND: TEMPERATURE INDEX
C                               2ND IND: RATIO INDEX
C
C
C PROGRAM: (I*4)  IT         = GENERAL INDEX FOR TEMPERATURE
C          (I*4)  IZ         = GENERAL INDEX FOR CHARGE
C          (I*4)  IP         = GENERAL INDEX FOR CHARGE
C          (I*4)  IZ1        = GENERAL INDEX FOR CHARGE+1
C          (I*4)  IL         = GENERAL INDEX FOR LINE
C          (I*4)  IR         = GENERAL INDEX FOR RATIO
C          (I*4)  ICPT       = GENERAL INDEX FOR LINE COMPONENT
C
C
C ROUTINES:
```

```
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C
C
C AUTHOR : H. P. SUMMERS, JET
C          K1/1/57
C          JET  EXT. 4941
C
C DATE   : 03/05/94
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 08-11-95
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST RELEASE
C
C VERSION: 1.2                          DATE: 08-11-95
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - REMOVED SUPERFLUOUS VARIABLES
C
C-----------------------------------------------------------------------
      CHARACTER           CIMET(NDLINE,NDCOMP)
      CHARACTER           CINDPH(NDLINE,NDCOMP)
      CHARACTER*16        SPECL(NDLINE,NDCOMP)
      INTEGER             IFILE(NDLINE,NDCOMP),     ILINE(NDRAT)
      INTEGER             IMDIMD,       IMET(NDLINE,NDCOMP)
      INTEGER             IMIP(IMDIMD),             INDPH(NDLINE,NDCOMP)
      INTEGER             IPDIMD,       IPIZM(IZDIMD,IPDIMD)
      INTEGER             IPLINE(NDLINE,NDCOMP),    ISDIMD,       ITDIMD
      INTEGER             ITMAX,        IZ0,          IZDIMD,       IZH
      INTEGER             IZION(NDLINE,NDCOMP),     IZIP(IMDIMD)
      INTEGER             IZL,          JLINE(NDRAT)
      INTEGER             NCOMP(NDLINE),            NDCOMP,       NDLINE
      INTEGER             NDRAT,        NLINE,        NMSUM,        NRAT
      INTEGER             NTDIM
      LOGICAL             LPEC(NDLINE,NDCOMP)
      REAL*8              DENS(NTDIM), DENSH(NTDIM)
      REAL*8              FPABUN(NTDIM,IMDIMD),     GCFEQ(NTDIM,NDLINE)
      REAL*8              GCFPEQ(NTDIM,NDLINE,NDCOMP)
      REAL*8              PECA(NTDIM,NDLINE,NDCOMP)
      REAL*8              RATA(NTDIM,NDRAT)
```

5.25 d5spc2: Subroutine d5spc2 from library adas4xx

```
      SUBROUTINE D5SPC2( DSNAME, IBSEL  , IZIN   , IZ0IN  ,
     &                   ITVAL  , TVAL   , DVAL   ,
     &                   WLNGTH ,
     &                   PECA   , LTRNG  , LDRNG  ,
     &                   TITLX  , IRCODE
     &                 )
C-----------------------------------------------------------------------
C
C  ******************* FORTRAN77 SUBROUTINE: D5PSC2 *********************
C
C  PURPOSE: TO EXTRACT AND INTERPOLATE  PHOTON EMISSIVITIES FOR
C           EMITTING IONS.
C
C     THIS IS A NEW ROUTINE, WRITTEN BECAUSE OF THE VERY
C           NAMING CONVENTIONS ON THE IBM AND UNIX MACHINES.
C           IT REPLACES THE OLD SPEC FORTRAN ROUTINE AND A LOT
C           OF THE OBSOLETE FUNCTIONALITY THEREIN. THIS ROUTINE
C           TAKES AS INPUT THE NAMES OF THE PHOTON EMISSIVITY FILES
C           AND CHECKS THEY ARE THERE BEFORE OPENING THEM AND
C           EXTRACTING ALL REQUIRED INFORMATION.
C
C  CALLING PROGRAM: D5SPEC
C
C  SUBROUTINE:
C
C  INPUT : (I*4)   IBSEL   = INDEX OF DATA-BLOCK SELECTED FOR ANALYSIS
C  INPUT : (I*4)   IZIN    = ION CHARGE OF EMITTING ION
C  INPUT : (I*4)   IZ0IN   = NUCLEAR CHARGE OF EMITTING ION
C
C  INPUT : (I*4)   ITVAL   = NO. OF ELECTRON TEMPERATURE/DENSITY PAIRS
C  INPUT : (R*8)   TVAL()  = ELECTRON TEMPERATURES (UNITS: EV)
C                            DIMENSION: TEMPERATURE/DENSITY PAIR INDEX
C  INPUT : (R*8)   DVAL()  = ELECTRON DENSITIES (UNITS: CM-3)
C                            DIMENSION: TEMPERATURE/DENSITY PAIR INDEX
C
C  OUTPUT: (R*8)   WLNGTH  = SELECTED BLOCK WAVELENGTH (ANGSTROMS)
C
C  OUTPUT: (R*8)   PECA()  = PHOTON EMISSIVITIES.
C                            DIMENSION: TEMPERATURE/DENSITY PAIR INDEX
C  OUTPUT: (L*4)   LTRNG() =.TRUE.  => OUTPUT 'PECA()'  VALUE WAS INTER-
C                                      POLATED  FOR  THE  USER  ENTERED
C                                      ELECTRON TEMPERATURE 'TVAL()'.
C                           .FALSE. => OUTPUT 'PECA()'  VALUE WAS EXTRA-
C                                      POLATED  FOR  THE  USER  ENTERED
C                                      ELECTRON TEMPERATURE 'TVAL()'.
C                            DIMENSION: TEMPERATURE/DENSITY PAIR INDEX
C  OUTPUT: (L*4)   LDRNG() =.TRUE.  => OUTPUT 'PECA()'  VALUE WAS INTER-
C                                      POLATED  FOR  THE  USER  ENTERED
C                                      ELECTRON DENSITY 'DVAL()'.
C                           .FALSE. => OUTPUT 'PECA()'  VALUE WAS EXTRA-
C                                      POLATED  FOR  THE  USER  ENTERED
C                                      ELECTRON DENSITY 'DVAL()'.
C                            DIMENSION: TEMPERATURE/DENSITY PAIR INDEX
C
C  OUTPUT: (C*120) TITLX   = INFORMATION STRING (DSN ETC.)
C  OUTPUT: (I*4)   IRCODE  = RETURN CODE FROM SUBROUTINE:
C                            0 => NORMAL COMPLETION - NO ERROR DETECTED
C                            1 => DATA SET MEMBER FOR EMITTING ION WITH
C                                 CHARGE 'IZIN' & ION CHARGE 'IZ0IN' CAN
```

```
C                                NOT BE FOUND/DOES NOT EXIST.
C                             2 => DISCREPANCY BETWEEN REQUESTED CHARGES
C                                  AND THOSE IN INPUT FILE.
C                             3 => THE SELECTED DATA-BLOCK 'IBSEL' IS OUT
C                                  OF RANGE OR DOES NOT EXIST.
C                             4 => INVALID VALUE FOR 'IZ0IN' ENTERED.
C                                  ('IZ0MIN' <= 'IZ0IN' <= 'IZ0MAX')
C                             5 => INVALID VALUE FOR 'IZIN' ENTERED.
C                                  ( 0  <= 'IZIN' <= 99 )
C                             9 => ERROR ENCOUNTERED WHEN TRYING TO OPEN
C                                  INPUT DATA-SET.
C
C          (I*4)   NSTORE  = PARAMETER= MAXIMUM NUMBER  OF  DATA-BLOCKS
C                                       WHICH CAN BE READ FROM THE INPUT
C                                       DATA-SET.
C          (I*4)   NTDIM   = PARAMETER= MAXIMUM NUMBER OF ELECTRON TEMP-
C                                       ERATURES THAT CAN BE READ  FROM
C                                       AN INPUT DATA-SET DATA-BLOCK.
C          (I*4)   NDDIM   = PARAMETER= MAXIMUM NUMBER OF ELECTRON DENS-
C                                       ITIES  THAT  CAN  BE  READ  FROM
C                                       AN INPUT DATA-SET DATA-BLOCK.
C          (I*4)   IZ0MIN  = PARAMETER: MIN. ALLOWED VALUE FOR 'IZ0IN'
C          (I*4)   IZ0MAX  = PARAMETER: MAX. ALLOWED VALUE FOR 'IZ0IN'
C
C          (I*4)   IUNIT   = UNIT TO WHICH INPUT DATA SET IS ALLOCATED
C          (I*4)   NBSEL   = TOTAL NUMBER OF DATA-BLOCKS READ FROM INPUT
C                            DATA SET.
C          (I*4)   IZ0     = INPUT FILE - EMITTING ION - NUCLEAR CHARGE
C          (I*4)   IZ      = INPUT FILE - EMITTING ION - CHARGE
C          (I*4)   IZ1     = INPUT FILE - EMITTING ION - CHARGE + 1
C
C          (L*4)   LOPEN   = .TRUE.  => INPUT DATA SET OPEN.
C                            .FALSE. => INPUT DATA SET CLOSED.
C
C          (C*2)   ESYM    = INPUT FILE - EMITTING ION - ELEMENT SYMBOL
CA         (C*120)  DSNAME  = NAME OF DATA SET INTERROGATED
C
C          (I*4)   ISELA() = INPUT DATA FILE: DATA-BLOCK ENTRY INDICES.
C                            DIMENSION: DATA-BLOCK INDEX
C          (I*4)   ITA()   = INPUT DATA SET-NUMBER OF ELECTRON TEMPERA-
C                            TURES.
C                            DIMENSION: DATA-BLOCK INDEX
C          (I*4)   IDA()   = INPUT DATA SET-NUMBER OF ELECTRON DENSITIES
C                            DIMENSION: DATA-BLOCK INDEX
C
C          (R*8)   TETA(,) = INPUT DATA SET -
C                            ELECTRON TEMPERATURES (UNITS: eV)
C                            1st DIMENSION: ELECTRON TEMPERATURE INDEX
C                            2nd DIMENSION: DATA-BLOCK INDEX
C          (R*8)   TEDA(,) = INPUT DATA SET -
C                            ELECTRON DENSITIES    (UNITS: cm-3)
C                            1st DIMENSION: ELECTRON DENSITY     INDEX
C                            2nd DIMENSION: DATA-BLOCK INDEX
C          (R*8)   PEC(,,)   =INPUT DATA SET -
C                             FULL SET OF IONIZATIONS PER PHOTON
C                             1st DIMENSION: ELECTRON TEMPERATURE INDEX
C                             2nd DIMENSION: ELECTRON DENSITY     INDEX
C                             3rd DIMENSION: DATA-BLOCK INDEX
C
C          (C*10)  CWAVEL() = INPUT FILE - WAVELENGTH (ANGSTROMS)
C                             DIMENSION: DATA-BLOCK INDEX
```

```
C          (C*8)   CFILE()  = INPUT FILE - SPECIFIC ION FILE SOURCE
C                             DIMENSION: DATA-BLOCK INDEX
C          (C*8)   CTYPE()  = INPUT FILE - TYPE OF DATA (IE EXCIT., ETC)
C                             DIMENSION: DATA-BLOCK INDEX
C          (C*2)   CINDM()  = INPUT FILE - METASTABLE INDEX
C                             DIMENSION: DATA-BLOCK INDEX
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          E3DATA     ADAS      FETCH INPUT DATA FROM SELECTED DATA SET
C          E3CHKB     ADAS      CHECK VALIDITY OF ION AND 'IBSEL'
C          E3SPLN     ADAS      INTERPOLATE DATA WITH TWO WAY SPLINES
C          E3TITL     ADAS      CREATE DESCRIPTIVE TITLE FOR OUTPUT
C
C AUTHOR:  TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C DATE:    08/11/95
C
C VERSION: 1.1 DATE: 08-11-95
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST RELEASE
C
C VERSION: 1.2 DATE: 10-11-95
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - INCREASED LENGTH OF TITLX FROM 80 TO 120
C
C VERSION: 1.3 DATE: 08-06-98
C MODIFIED: RICHARD MARTIN
C     - INCREASED NTDIM AND NDDIM TO 30.
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      CHARACTER*120       DSNAME,      TITLX
      INTEGER             IBSEL,       IRCODE,      ITVAL,       IZ0IN
      INTEGER             IZIN
      LOGICAL             LDRNG(ITVAL),             LTRNG(ITVAL)
      REAL*8              DVAL(ITVAL), PECA(ITVAL), TVAL(ITVAL), WLNGTH
```

5.26 d5spec: Subroutine d5spec from library adas4xx

```
       SUBROUTINE D5SPEC( LRSPEC ,
     &                    NDLINE , NDCOMP  , NDRAT  , NDFILE ,
     &                    NFILE  , LFILE   ,
     &                    UID    , GROUP   , TYPE   , EXT    ,
     &                    IZ0    , DSPECA  ,
     &                    NLINE  , NCOMP   ,
     &                    IZION  , IMET    , CIMET  , INDPH  ,
     &                    IFILE  ,
     &                    NTDIM  , ITMAX  ,
     &                    TEIN   , DEIN    , THIN   , DHIN   ,
     &                    PECA   ,
     &                    LPEC   , LTRNG   , LDRNG
     &                  )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: D5SPEC  *******************
C
C  PURPOSE: TO CALCULATE PHOTON EMISSIVITY COEFFICIENTS FOR
C           SPECTRAL LINES IDENTIFIED IN SCRIPT FILE
C
C  CALLING PROGRAM: ADAS405
C
C  SUBROUTINE:
C
C  INPUT : (I*4)   NDLINE   = MAXIMUM NUMBER OF LINES ALLOWED
C  INPUT : (I*4)   NDCOMP   = MAXIMUM NUMBER OF COMPONENT FOR EACH LINE
C  INPUT : (I*4)   NDRAT    = MAXIMUM NUMBER OF LINE RATIOS ALLOWED
C  INPUT : (I*4)   NDFILE   = MAXIMUM NUMBER OF EMISSIVITY FILES WHICH
C                             CAN BE SEARCHED
C  INPUT : (I*4)   NFILE    = NUMBER OF PEC FILES TO BE SCANNED
C  INPUT : (L*4)   LFILE()  = .TRUE.  => PEC FILE EXISTS AND MATCHES
C                             .FALSE. => PEC FILE DOES NOT EXIST/MATCH
C  INPUT : (C*6)   UID()    = USER IDENTIFIER OF PEC FILE
C  INPUT : (C*8)   GROUP()  = GROUP IDENTIFIER OF PEC FILE
C  INPUT : (C*5)   TYPE()   = TYPE IDENTIFIER OF PEC FILE
C  INPUT : (C*3)   EXT()    = EXTENSION OF PEC FILE MEMBER NAME
C  INPUT : (I*4)   IZ0      = NUCLEAR CHARGE OF IMPURITY
C  INPUT : (C*120)DSPECA()  = PHOTON EMISSIVITY SOURCE FILES
C  INPUT : (I*4)   NLINE    = NUMBER OF LINES IDENTIFIED IN SCRIPT
C  INPUT : (I*4)   NCOMP()  = NUMBER OF COMPONENTS OF SCRIPT LINE
C                             1ST DIM: LINE INDEX
C  INPUT : (I*4)   IZION(,) = CHARGE STATE OF COMPONENT
C                             1ST DIM: LINE INDEX
C                             2ND DIM: COMPONENT INDEX
C  INPUT : (I*4)   IMET(,)  = NUMBER OF COMPONENTS OF SCRIPT LINE
C                             1ST DIM: LINE INDEX
C                             2ND DIM: COMPONENT INDEX
C  INPUT : (C*1)   CIMET(,) = SIGN (+, BLANK OR -) OF METASTABLE
C                             1ST DIM: LINE INDEX
C                             2ND DIM: COMPONENT INDEX
C  INPUT : (I*4)   INDPH(,) = PEC FILE INDEX OF LINE COMPONENT
C                             1ST DIM: LINE INDEX
C                             2ND DIM: COMPONENT INDEX
C  INPUT : (I*4)   IFILE(,) = INDEX OF PEC FILE IN FILE LIST
C                             1ST DIM: LINE INDEX
C                             2ND DIM: COMPONENT INDEX
C  INPUT : (I*4)   NTDIM    = MAXIMUM NUMBER OF TEMP/DENSITY SETS
C  INPUT : (I*4)   ITMAX    = NUMBER OF TEMP/DENSITY SETS
C  INPUT : (R*8)   TEIN()   = ELECTRON TEMPERATURES (EV)
```

```
C  INPUT : (R*8)   DEIN()   = ELECTRON DENSITIES (CM-3)
C  INPUT : (R*8)   THIN()   = HYDROGEN TEMPERATURES (EV)
C  INPUT : (R*8)   DHIN()   = HYDROGEN DENSITIES (CM-3)
C
C  OUTPUT: (L*4)   LRSPEC   = .TRUE.  => PEC PROCESSING DONE
C                             .FALSE. => PEC PROCESSING NOT DONE
C  OUTPUT: (R*8)   PECA(,,) = PHOTON EMISSIVITY COEFFICIENTS (CM3 S-1)
C                             1ST DIM: TEMPERATURE INDEX
C                             2ND DIM: LINE INDEX
C                             3RD DIM: COMPONENT INDEX
C  OUTPUT: (L*4)   LPEC(,)  = .TRUE.  => PHOTON EMISSIVITY OBTAINED
C                             .FALSE. => PHOTON EMISSIVITY NOT OBTAINED
C                             2ND DIM: LINE INDEX
C                             3RD DIM: COMPONENT INDEX
C
C          (I*4)   IUNT10  = PARAMETER = INPUT UNIT FOR DATA
C          (L*4)   OPEN10  = .TRUE.  => FILE ALLOCATED TO UNIT 10.
C                            .FALSE. => NO FILE ALLOCATED TO UNIT 10.
C
C ROUTINES:
C          ROUTINE   SOURCE   BRIEF DESCRIPTION
C          ------------------------------------------------------------
C    D5SPC2   IDL-ADAS  OBTAIN PHOTON EMISSIVITY COEFFICIENT
C
C AUTHOR:  H. P. SUMMERS, JET
C          K1/1/57
C          JET EXT. 4941
C
C DATE:    20/04/94
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 08-11-95
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST RELEASE
C
C VERSION: 1.2                          DATE: 08-11-95
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - REMOVED SUPERFLUOUS VARIABLES
C
C VERSION: 1.3                          DATE: 10-11-95
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - INCREASED LENGTH OF TITLX FROM 80 TO 120
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      CHARACTER           CIMET(NDLINE,NDCOMP)
      CHARACTER*120       DSPECA(NDFILE)
      CHARACTER*3         EXT(NDFILE)
      CHARACTER*8         GROUP(NDFILE)
      CHARACTER*5         TYPE(NDFILE)
      CHARACTER*6         UID(NDFILE)
      INTEGER             IFILE(NDLINE,NDCOMP),     IMET(NDLINE,NDCOMP)
      INTEGER             INDPH(NDLINE,NDCOMP),     ITMAX,       IZ0
      INTEGER             IZION(NDLINE,NDCOMP),     NCOMP(NDLINE)
      INTEGER             NDCOMP,      NDFILE,      NDLINE,      NDRAT
      INTEGER             NFILE,       NLINE,       NTDIM
      LOGICAL             LDRNG(NTDIM),             LFILE(NDFILE)
      LOGICAL             LPEC(NDLINE,NDCOMP),      LRSPEC
      LOGICAL             LTRNG(NTDIM)
```

```
REAL*8              DEIN(NTDIM), DHIN(NTDIM)
REAL*8              PECA(NTDIM,NDLINE,NDCOMP)
REAL*8              TEIN(NTDIM), THIN(NTDIM)
```

5.27 d5spow: Subroutine d5spow from library adas4xx

```
       SUBROUTINE D5SPOW( LSELA  , LEXSA  , LDEFA  , LPART  , LEXSS  ,
     &                    IZ0    , IZL    , IZH    , NPART  ,
     &                    ISDIMD , IZDIMD , ITDIMD , IPDIMD , IMDIMD ,
     &                    ACDA   , SCDA   , CCDA   , PRBA   ,
     &                    PRCA   , QCDA   , XCDA   , PLTA   ,
     &                    NMSUM  , IZIP   , IMIP   , IPIZM  ,
     &                    NTDIM  , ITMAX  ,
     &                    DENS   , DENSH  ,
     &                    FPABUN , FSABUN ,
     &                    PLTPEQ ,
     &                    ACDSEQ , SCDSEQ , CCDSEQ , PRBSEQ ,
     &                    PRCSEQ , PLTSEQ ,
     &                    PRBEQ  , PRCEQ  , PLTEQ  , PRADA
     &                  )
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: D5SPOW *********************
C
C PURPOSE : TO ASSEMBLE RADIATED POWER FUNCTIONS USING FRACTIONAL
C           METASTABLE ABUNDANCES.
C           GENERATE STANDARD ISONUCLEAR MASTER DATA FROM PARTIAL DATA.
C
C NOTE    : THE SOURCE ISONUCLEAR MASTER FILE DATA ARE OBTAINED BY A
C           PRIOR CALL TO SUBROUTINE D5DATA FROM SEQUENTIAL FILES
C           WITH THE FOLLOWING NAMING CONVENTIONS:
C
C                   (1) JETSHP.ACD<YR>#<EL).<CODE>DATA
C                   (2) JETSHP.SCD<YR>#<EL>.<CODE>DATA
C                   (3) JETSHP.CCD<YR>#<EL>.<CODE>DATA
C                   (4) JETSHP.PRB<YR>#<EL>.<FILT>.<CODE>DATA
C                   (5) JETSHP.PRC<YR>#<EL>.<FILT>.<CODE>DATA
C                   (6) JETSHP.QCD<YR>#<EL>.<CODE>DATA
C                   (7) JETSHP.XCD<YR>#<EL>.<CODE>DATA
C                   (8) JETSHP.PLT<YR>#<EL>.<CODE>DATA
C
C       WHERE, <YR>   = TWO DIGIT YEAR NUMBER
C              <EL>   = ONE OR TWO CHARACTER ELEMENT SYMBOL
C              <CODE> = R       => PARTIAL DATA
C                       U       => PARTIAL DATA
C                       OMITTED => STANDARD DATA
C              <FILT> = SIX CHARACTER POWER FILTER CODE
C
C       AND DATA OF CLASSES 6 AND 7 DO NOT EXIST FOR THE PARTIAL CASE.
C
C
C INPUT  : (L*4)  LSELA()   = .TRUE.  => INPUT DATA SET TYPE FOR THIS
C                                        INDEX SELECTED
C                           = .FALSE. => INPUT DATA SET FOR THIS INDEX
C                                        NOT SELECTED
C INPUT  : (L*4)  LEXSA()   = .TRUE.  => INPUT DATA SET TYPE FOR THIS
C                                        SELECTED INDEX EXISTS
C                           = .FALSE. => INPUT DATA SET DOES NOT EXIST
C                                        FOR THIS SELECTED INDEX
C INPUT  : (L*4)  LDEFA()   = .TRUE.  => INPUT DATA SET TYPE FOR THIS
C                                        DEFAULT YEAR INDEX EXISTS
C                           = .FALSE. => INPUT DATA SET DOES NOT EXIST
C                                        FOR THIS DEFAULT YEAR INDEX
C INPUT  : (L*4)  LPART     = .TRUE.  => PARTIAL DATA SELECTED
```

```
C                             = .FALSE. => STANDARD DATA SELECTED
C INPUT  : (I*4)  IZ0       = NUCLEAR CHARGE
C INPUT  : (I*4)  IZL       = MINIMUM ION CHARGE+1 IN MASTER DATA FILES
C INPUT  : (I*4)  IZH       = MAXIMUM ION CHARGE+1 IN MASTER DATA FILES
C INPUT  : (I*4)  NPART()   = METASTABLE PARTITION.  I.E. NUMBER OF
C                             METASTABLES FROM CHARGE STATE IZL-1 TO
C                             IZH ON INPUT
C INPUT  : (I*4)  ISDIMD    = MAXIMUM NUMBER OF (CHARGE, PARENT, GROUND)
C                             BLOCKS IN ISONUCLEAR MASTER FILES
C INPUT  : (I*4)  IZDIMD    = MAXIMUM NUMBER OF CHARGE STATES
C                             IN ISONUCLEAR MASTER FILES
C INPUT  : (I*4)  ITDIMD    = MAXIMUM NUMBER OF TEMP OR DENS VALUES IN
C                             ISOELECTRONIC MASTER FILES
C INPUT  : (I*4)  IPDIMD    = MAXIMUM NUMBER OF METASTABLES FOR EACH
C                             IONISATION STAGE
C INPUT  : (I*4)  IMDIMD    = MAXIMUM NUMBER OF METASTABLES
C
C INPUT  : (R*8)  ACDA(,,,) = INTERPOLATION OF ACD COEFFICIENT (CM3 S-1)
C                             1ST DIM: TEMPERATURE INDEX
C                             2ND DIM: CHARGE STATE INDEX
C                             3RD DIM: RECOMBINING METASTABLE INDEX
C                             4TH DIM: RECOMBINED METASTABLE INDEX
C INPUT  : (R*8)  SCDA(,,,) = INTERPOLATION OF SCD COEFFICIENT (CM3 S-1)
C                             1ST DIM: TEMPERATURE INDEX
C                             2ND DIM: CHARGE STATE INDEX
C                             3RD DIM: RECOMBINING METASTABLE INDEX
C                             4TH DIM: RECOMBINED METASTABLE INDEX
C INPUT  : (R*8)  CCDA(,,,) = INTERPOLATION OF CCD COEFFICIENT (CM3 S-1)
C                             1ST DIM: TEMPERATURE INDEX
C                             2ND DIM: CHARGE STATE INDEX
C                             3RD DIM: RECOMBINING METASTABLE INDEX
C                             4TH DIM: RECOMBINED METASTABLE INDEX
C INPUT  : (R*8)  PRBA(,,)  = INTERPOLATION OF PRB COEFFICIENT (W CM3 )
C                             1ST DIM: TEMPERATURE INDEX
C                             2ND DIM: CHARGE STATE INDEX
C                             3RD DIM: RECOMBINING METASTABLE INDEX
C INPUT  : (R*8)  PRCA(,,)  = INTERPOLATION OF PRC COEFFICIENT (W CM3 )
C                             1ST DIM: TEMPERATURE INDEX
C                             2ND DIM: CHARGE STATE INDEX
C                             3RD DIM: RECOMBINING METASTABLE INDEX
C INPUTT : (R*8)  QCDA(,,,) = INTERPOLATION OF QCD COEFFICIENT (CM3 S-1)
C                             1ST DIM: TEMPERATURE INDEX
C                             2ND DIM: CHARGE STATE INDEX
C                             3RD DIM: FIRST METASTABLE INDEX
C                             4TH DIM: SECOND METASTABLE INDEX
C INPUT  : (R*8)  XCDA(,,,) = INTERPOLATION OF XCD COEFFICIENT (CM3 S-1)
C                             1ST DIM: TEMPERATURE INDEX
C                             2ND DIM: CHARGE STATE INDEX
C                             3RD DIM: FIRST PARENT METASTABLE INDEX
C                             4TH DIM: SECOND PARENT METASTABLE INDEX
C INPUT  : (R*8)  PLTA(,,)  = INTERPOLATION OF PLT COEFFICIENT (W CM3 )
C                             1ST DIM: TEMPERATURE INDEX
C                             2ND DIM: CHARGE STATE INDEX
C                             3RD DIM: METASTABLE INDEX
C INPUT  : (I*4)  NMSUM     = TOTAL NUMBER OF POPULATIONS
C
C INPUT  :        IZIP()    = ION CHARGE +1 (IZ1) OF METASTABLE IN LIST
C INPUT  :        IMIP()    = METASTABLE INDEX WITHIN CHARGE STATE IZ1
C                             OF METASTABLE INDEX FROM COMPLETE LIST
C INPUT  :        IPIZM(,)  = METASTABLE INDEX IN COMPLETE LIST
C                             1ST DIM: INDEX IZ1-IZL+1
```

```
C                               2ND DIM: METASTABLE COUNT FOR STAGE (IGRD)
C INPUT  : (I*4)  NTDIM     = MAXIMUM NUMBER OF DTEV/DDENS PAIRS
C INPUT  : (I*4)  ITMAX     = NUMBER OF ( DTEV() , DDENS() ) PAIRS
C INPUT  : (R*8)  DENS()    = ELECTRON DENSITIES (CM-3))
C INPUT  : (R*8)  DENSH()   = HYDROGEN DENSITIES (CM-3))
C INPUT  : (R*8)  FPABUN(,) = RESOLVED METASTABLE EQUILIBRIUM
C                             FRACTIONAL ABUNDANCES
C                             1ST DIM: - TEMPERATURE/DENSITY PAIR
C                             2ND DIM: - METASTABLE INDEX
C OUTPUT : (L*4)  LEXSS()   = .TRUE.  => OUTPUT STANDARD MASTER DATA FOR
C                                        THIS INDEX GENERATED
C                           = .FALSE. => OUTPUT STANDARD MASTER DATA FOR
C                                        THIS INDEX NOT GENERATED
C OUTPUT : (R*8)  FSABUN(,) = STAGE EQUILIBRIUM FRACTIONAL ABUNDANCES
C                             1ST DIM: - TEMPERATURE/DENSITY PAIR INDEX
C                             2ND DIM: - CHARGE STATE INDEX (IZ1-IZL+1)
C OUTPUT : (R*8)  PLTPEQ(,) = METASTABLE PARTIAL EQUILIBRIUM RADIATED
C                             LINE POWER FUNCTIONS
C                             1ST DIM: - TEMPERATURE/DENSITY PAIR
C                             2ND DIM: - METASTABLE INDEX
C OUTPUT : (R*8)  ACDSEQ(,) = STANDARD (UNRESOLVED) ACD COEFFICIENT
C                             1ST DIM: - TEMPERATURE/DENSITY PAIR INDEX
C                             2ND DIM: - CHARGE STATE INDEX (IZ1-IZL+1)
C OUTPUT : (R*8)  SCDSEQ(,) = STANDARD (UNRESOLVED) SCD COEFFICIENT
C                             1ST DIM: - TEMPERATURE/DENSITY PAIR INDEX
C                             2ND DIM: - CHARGE STATE INDEX (IZ1-IZL+1)
C OUTPUT : (R*8)  CCDSEQ(,) = STANDARD (UNRESOLVED) CCD COEFFICIENT
C                             1ST DIM: - TEMPERATURE/DENSITY PAIR INDEX
C                             2ND DIM: - CHARGE STATE INDEX (IZ1-IZL+1)
C OUTPUT : (R*8)  PRBSEQ(,) = STANDARD (UNRESOLVED) SCD COEFFICIENT
C                             1ST DIM: - TEMPERATURE/DENSITY PAIR INDEX
C                             2ND DIM: - CHARGE STATE INDEX (IZ1-IZL+1)
C OUTPUT : (R*8)  PRCSEQ(,) = STANDARD (UNRESOLVED) CCD COEFFICIENT
C                             1ST DIM: - TEMPERATURE/DENSITY PAIR INDEX
C                             2ND DIM: - CHARGE STATE INDEX (IZ1-IZL+1)
C OUTPUT : (R*8)  PLTSEQ(,) = STANDARD (UNRESOLVED) CCD COEFFICIENT
C                             1ST DIM: - TEMPERATURE/DENSITY PAIR INDEX
C                             2ND DIM: - CHARGE STATE INDEX (IZ1-IZL+1)
C OUTPUT : (R*8)  PRBEQ()   = TOTAL EQUILIBRIUM RADIATED RECOM-BREMS
C                                   POWER FUNCTION
C OUTPUT : (R*8)  PRCEQ()   = TOTAL EQUILIBRIUM CX RADIATED RECOM POWER
C                                   FUNCTION NORMALISED TO ELECTRON
C                                   DENSITY
C OUTPUT : (R*8)  PLTEQ()   = TOTAL EQUILIBRIUM RADIATED LINE POWER
C                                   FUNCTION
C OUTPUT : (R*8)  PRADA()   = TOTAL EQUILIBRIUM RADIATED POWER FUNCTION
C
C PROGRAM: (I*4)  IT        = GENERAL INDEX FOR TEMPERATURE
C          (I*4)  IZ        = GENERAL INDEX FOR CHARGE
C          (I*4)  IP        = GENERAL INDEX FOR CHARGE
C          (I*4)  IZ1       = GENERAL INDEX FOR CHARGE+1
C          (I*4)  ICL       = GENERAL INDEX FOR CLASS
C          (I*4)  IPP       = GENERAL PARENT INDEX
C          (I*4)  IPG       = GENERAL GROUND INDEX
C          (I*4)  IZREF     = GENERAL CHARGE STAE POINTER INDEX
C          (I*4)  IPRT      = GENERAL INDEX FOR PARENT METASTABLE
C          (I*4)  IGRD      = GENERAL INDEX FOR METASTABLE
C
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
```

```
C          ------------------------------------------------------------
C
C
C AUTHOR : H. P. SUMMERS, JET
C          K1/1/57
C          JET  EXT. 4941
C
C DATE   : 28/04/94
C
C UNIX-IDL PORT:
C VERSION:     1.1                          DATE: 31-10-95
C MODIFIED:    TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C              - INITIAL VERSION TO BE USED FOR UNIX PLATFORMS
C
C VERSION:     1.2                          DATE: 08-11-95
C MODIFIED:    TIM HAMMOND
C              - ALTERED LINE 'IF(LEXSA(ICL).OR.LDEFA(ICL))' IN LOOP 1
C                TO 'IF((LEXSA(ICL).OR.LDEFA(ICL)).AND.LSELA(ICL))' TO
C                REFLECT WHETHER OR NOT THE USER HAS ACTUALLY SELECTED
C                THIS PARTICULAR CLASS FOR INCLUSION.
C              - TIDIED UP COMMENTS AND CODE
C
C VERSION:     1.3                          DATE: 08-11-95
C MODIFIED:    TIM HAMMOND
C              - REMOVED SUPERFLUOUS VARIABLES
C
C-----------------------------------------------------------------------
      INTEGER             IMDIMD,      IMIP(IMDIMD),             IPDIMD
      INTEGER             IPIZM(IZDIMD,IPDIMD),     ISDIMD,      ITDIMD
      INTEGER             ITMAX,       IZ0,         IZDIMD,      IZH
      INTEGER             IZIP(IMDIMD),             IZL,         NMSUM
      INTEGER             NPART(IZDIMD),            NTDIM
      LOGICAL             LDEFA(8),    LEXSA(8),    LEXSS(8),    LPART
      LOGICAL             LSELA(8)
      REAL*8              ACDA(NTDIM,IZDIMD,IPDIMD,IPDIMD)
      REAL*8              ACDSEQ(NTDIM,IZDIMD)
      REAL*8              CCDA(NTDIM,IZDIMD,IPDIMD,IPDIMD)
      REAL*8              CCDSEQ(NTDIM,IZDIMD),     DENS(NTDIM)
      REAL*8              DENSH(NTDIM),             FPABUN(NTDIM,IMDIMD)
      REAL*8              FSABUN(NTDIM,IZDIMD)
      REAL*8              PLTA(NTDIM,IZDIMD,IPDIMD)
      REAL*8              PLTEQ(NTDIM),             PLTPEQ(NTDIM,IMDIMD)
      REAL*8              PLTSEQ(NTDIM,IZDIMD),     PRADA(NTDIM)
      REAL*8              PRBA(NTDIM,IZDIMD,IPDIMD)
      REAL*8              PRBEQ(NTDIM),             PRBSEQ(NTDIM,IZDIMD)
      REAL*8              PRCA(NTDIM,IZDIMD,IPDIMD)
      REAL*8              PRCEQ(NTDIM),             PRCSEQ(NTDIM,IZDIMD)
      REAL*8              QCDA(NTDIM,IZDIMD,IPDIMD,IPDIMD)
      REAL*8              SCDA(NTDIM,IZDIMD,IPDIMD,IPDIMD)
      REAL*8              SCDSEQ(NTDIM,IZDIMD)
      REAL*8              XCDA(NTDIM,IZDIMD,IPDIMD,IPDIMD)
```

5.28 d5wr11: Subroutine d5wr11 from library adas4xx

```
      SUBROUTINE D5WR11( IUNIT  , DSNINC , DSNMTR , ELEMT ,
     &                   UID    , DATE   , IZ0    ,
     &                   NDLINE , NLINE  ,
     &                   TITL   , IZION  , CIMET  ,
     &                   NTDIM  , ITMAX  ,
     &                   TEV    , DENS   ,
     &                   GCF
     &                 )
C-----------------------------------------------------------------------
C
C  ******************** FORTRAN77 SUBROUTINE: D5WR11 ******************
C
C  PURPOSE:  TO OUTPUT DATA TO GCF PASSING FILE.
C
C  CALLING PROGRAM: ADAS405
C
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  IUNIT   = OUTPUT UNIT NUMBER FOR RESULTS
C  INPUT : (C*80) DSNINC  = INPUT SCRIPT DATA SET NAME
C  INPUT : (C*80) DSNMTR  = INPUT ACD MASTER FILE NAME
C  INPUT : (C*2)  ELEMT   = ELEMENT SYMBOL.
C
C  INPUT : (C*80) UID     = USER IDENIFIER
C  INPUT : (C*8)  DATE    = CURRENT DATE
C  INPUT : (I*4)  IZ0     = NUCLEAR CHARGE
C
C  INPUT : (I*4)  NDLINE  = MAXIMUM NUMBER OF SPECTRUM LINES
C  INPUT : (I*4)  NLINE   = NUMBER OF SPECTRUM LINES
C  INPUT : (C*12) TITL()  = TITLE FOR 1ST COMPONENT OF LINE
C  INPUT : (I*4)  IZION() = ION CHARGE FOR 1ST COMPONENT OF LINE
C  INPUT : (C*1)  CIMET() = +/- SHIFT OF ION CHARGE - 1ST COMPONENT
C
C  INPUT : (I*4)  NTDIM   = AMXIMU, NUMBER OF INPUT TEMPERATURES
C  INPUT : (I*4)  ITMAX   = NUMBER OF INPUT TEMPERATURES
C                           LIST.
C  INPUT : (R*8)  TEV()   = ELECTRON TEMPERATURES (UNITS: EV)
C  INPUT : (R*8)  DENS()  = ELECTRON DENSITIES  (UNITS: CM-3)
C
C  INPUT : (R*4)  GCF(,)  = G(TE) FUNCTION (CM3 S-1)
C                             1ST DIM: TEMPERATURE INDEX
C                             2ND DIM: LINE INDEX
C
C          (I*4) I         = GENERAL USE
C          (I*4) J         = GENERAL USE
C          (I*4) K         = GENERAL USE
C          (I*4) L         = GENERAL USE
C          (I*4) I1        = GENERAL USE
C          (I*4) I2        = GENERAL USE
C          (I*4) IT        = GENERAL USE
C          (C*80)CLINE     = GENERAL USE
C
C
C ROUTINES: NONE
C
C AUTHOR:  H. P. SUMMERS, JET
C          K1/1/57
C          JET EXT. 4941
```

```
C
C DATE:    27/04/94
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 08-11-95
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST RELEASE
C
C VERSION: 1.2                          DATE: 08-11-95
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - REMOVED SUPERFLUOUS VARIABLES
C
C VERSION: 1.3                          DATE: 09-03-98
C MODIFIED: RICHARD MARTIN
C               - INCREASED UID FROM CHAR*6 TO CHAR*80 IN ACCORDANCE WITH
C XXUID.
C
C VERSION: 1.4                          DATE: 20-11-98
C MODIFIED: RICHARD MARTIN & MARTIN O'MULLANE
C     - REMOVED SEARCH FOR BRACKETS IN DSNINC AND INITIAL STRING
C AS STRING = '        '
C
C-----------------------------------------------------------------------
      CHARACTER           CIMET(NLINE)
      CHARACTER*8         DATE
      CHARACTER*80        DSNINC,      DSNMTR
      CHARACTER*2         ELEMT
      CHARACTER*12        TITL(NLINE)
      CHARACTER*80        UID
      INTEGER             ITMAX,       IUNIT,       IZ0
      INTEGER             IZION(NLINE),             NDLINE,      NLINE
      INTEGER             NTDIM
      REAL*8              DENS(NTDIM), GCF(NTDIM,NDLINE)
      REAL*8              TEV(NTDIM)
```

5.29 d6data: Subroutine d6data from library adas4xx

```
       SUBROUTINE D6DATA( DSFLLA , LSELA  , LEXSA  , LDEFA , LPART  ,
     &                    IZ0    , IZ1MIN , IZ1MAX , NPART  ,
     &                    NTDIM  , ITMAX  ,
     &                    ISDIMD , IZDIMD , ITDIMD , IPDIMD , NPARTR,
     &                    DTEV   , DDENS  ,
     &                    DTEVD  , DDENSD , DRCOFD , ZDATA  ,
     &                    DRCOFI ,
     &                    ACDA   , LACDA  ,
     &                    SCDA   , LSCDA  ,
     &                    CCDA   , LCCDA  ,
     &                    PRBA   , LPRBA  ,
     &                    PRCA   , LPRCA  ,
     &                    QCDA   , LQCDA  ,
     &                    XCDA   , LXCDA  ,
     &                    PLTA   , LPLTA
     &                  )
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: D6DATA *********************
C
C PURPOSE : TO EXTRACT A COMPLETE SET OF COLLISIONAL DIELECTRONIC DATA
C           FOR A TEMP/DENSITY MODEL
C           FROM EITHER PARTIAL (METASTABLE/PARENT RESOLVED) OR STANDARD
C           (UNRESOLVED) ISONUCLEAR MASTER FILES
C
C NOTE    : THE SOURCE DATA IS CONTAINED AS SEQUENTIAL DATASETS
C           WITH THE FOLLOWING NAMING CONVENTIONS:
C
C                   (1) JETSHP.ACD<YR>#<EL).<CODE>DATA
C                   (2) JETSHP.SCD<YR>#<EL>.<CODE>DATA
C                   (3) JETSHP.CCD<YR>#<EL>.<CODE>DATA
C                   (4) JETSHP.PRB<YR>#<EL>.<FILT>.<CODE>DATA
C                   (5) JETSHP.PRC<YR>#<EL>.<FILT>.<CODE>DATA
C                   (6) JETSHP.QCD<YR>#<EL>.<CODE>DATA
C                   (7) JETSHP.XCD<YR>#<EL>.<CODE>DATA
C                   (8) JETSHP.PLT<YR>#<EL>.<CODE>DATA
C
C       WHERE, <YR>   = TWO DIGIT YEAR NUMBER
C              <EL>   = ONE OR TWO CHARACTER ELEMENT SYMBOL
C              <CODE> = R       => PARTIAL DATA
C                       U       => PARTIAL DATA
C                       OMITTED => STANDARD DATA
C              <FILT> = SIX CHARACTER POWER FILTER CODE
C
C       AND DATA OF CLASSES 6 AND 7 DO NOT EXIST FOR THE STANDARD CASE.
C
C
C INPUT  : (C*120)DSFLLA()  = MASTER FILE DATA SET NAMES
C INPUT  : (L*4)  LSELA()   = .TRUE.  => INPUT DATA SET TYPE FOR THIS
C                                        INDEX SELECTED
C                           = .FALSE. => INPUT DATA SET FOR THIS INDEX
C                                        NOT SELECTED
C INPUT  : (L*4)  LEXSA()   = .TRUE.  => INPUT DATA SET TYPE FOR THIS
C                                        SELECTED INDEX EXISTS
C                           = .FALSE. => INPUT DATA SET DOES NOT EXIST
C                                        FOR THIS SELECTED INDEX
C INPUT  : (L*4)  LDEFA()   = .TRUE.  => INPUT DATA SET TYPE FOR THIS
C                                        DEFAULT YEAR INDEX EXISTS
```

```
C                             = .FALSE. => INPUT DATA SET DOES NOT EXIST
C                                          FOR THIS DEFAULT YEAR INDEX
C INPUT  : (L*4)  LPART     = .TRUE.  => PARTIAL DATA SELECTED
C                             = .FALSE. => STANDARD DATA SELECTED
C INPUT  : (I*4)  IZ0       = NUCLEAR CHARGE
C INPUT  : (I*4)  IZ1MIN    = MINIMUM ION CHARGE+1 IN MASTER DATA FILES
C INPUT  : (I*4)  IZ1MAX    = MAXIMUM ION CHARGE+1 IN MASTER DATA FILES
C INPUT  : (I*4)  NPART()   = METASTABLE PARTITION.  I.E. NUMBER OF
C                             METASTABLES FROM CHARGE STATE IZ1MIN-1 TO
C                             IZ1MAX ON INPUT
C INPUT  : (I*4)  NTDIM     = MAXIMUM NUMBER OF DTEV/DDENS PAIRS
C INPUT  : (I*4)  ITMAX     = NUMBER OF ( DTEV() , DDENS() ) PAIRS
C INPUT  : (I*4)  ISDIMD    = MAXIMUM NUMBER OF (CHARGE, PARENT, GROUND)
C                             BLOCKS IN ISONUCLEAR MASTER FILES
C INPUT  : (I*4)  IZDIMD    = MAXIMUM NUMBER OF CHARGE STATES
C                             IN ISONUCLEAR MASTER FILES
C INPUT  : (I*4)  ITDIMD    = MAXIMUM NUMBER OF TEMP OR DENS VALUES IN
C                             ISOELECTRONIC MASTER FILES
C INPUT  : (I*4)  IPDIMD    = MAXIMUM NUMBER OF METASTABLES FOR EACH
C                             IONISATION STAGE
C INPUT  : (R*8)  DTEV()    = DLOG10(ELECTRON TEMPERATURES (EV))
C INPUT  : (R*8)  DDENS()   = DLOG10(ELECTRON DENSITIES (CM-3))
C
C OUTPUT : (I*4)  NPARTR()  = METASTABLE PARTITION.  I.E. NUMBER OF
C                             METASTABLES FROM CHARGE STATE IZ1MIN-1 TO
C                             IZ1MAX FOUND IN MASTER FILE
C OUTPUT : (R*8)  DTEVD()   = DLOG10(DATA ELECTRON TEMPERATURES (EV))
C                             IN SELECTED MASTER FILE
C OUTPUT : (R*8)  DDENSD()  = DLOG10(DATA ELECTRON DENSITIES (CM-3))
C                             IN SELECTED MASTER FILE
C OUTPUT : (R*8)  DRCOFD(,,)= DLOG10(DATA RATE COEFFICIENTS (CM-3/S))
C                             IN SELECTED MASTER FILE
C                             1ST DIM: (CHARGE,META,GRD) BLOCK INDEX
C                             2ND DIM: TEMPERATURE INDEX
C                             3RD DIM: DENSITY INDEX
C OUTPUT : (R*8)  ZDATA()   = CHARGE + 1 FOR IONS IN SELECTED MASTER
C                             FILE
C                             1ST DIM: (CHARGE,META,GRD) BLOCK INDEX
C OUTPUT : (R*8)  DRCOFI()  = INTERPOLATION OF DRCOFD(,,) FOR
C                             DTEV() & DDENS()
C OUTPUT : (R*8)  ACDA(,,,) = INTERPOLATION OF ACD COEFFICIENT (CM3 S-1)
C                             1ST DIM: TEMPERATURE INDEX
C                             2ND DIM: CHARGE STATE INDEX
C                             3RD DIM: RECOMBINING METASTABLE INDEX
C                             4TH DIM: RECOMBINED METASTABLE INDEX
C OUTPUT : (L*4)  LACDA(,,) = .TRUE.  => ACD COEFFICIENT AVAILABLE
C                             .FALSE. => ACD COEFFICIENT NOT AVAILABLE
C                             1ST DIM: CHARGE STATE INDEX
C                             2ND DIM: RECOMBINING METASTABLE INDEX
C                             3RD DIM: RECOMBINED METASTABLE INDEX
C OUTPUT : (R*8)  SCDA(,,,) = INTERPOLATION OF SCD COEFFICIENT (CM3 S-1)
C                             1ST DIM: TEMPERATURE INDEX
C                             2ND DIM: CHARGE STATE INDEX
C                             3RD DIM: RECOMBINING METASTABLE INDEX
C                             4TH DIM: RECOMBINED METASTABLE INDEX
C OUTPUT : (L*4)  LSCDA(,,) = .TRUE.  => SCD COEFFICIENT AVAILABLE
C                             .FALSE. => SCD COEFFICIENT NOT AVAILABLE
C                             1ST DIM: CHARGE STATE INDEX
C                             2ND DIM: RECOMBINING METASTABLE INDEX
C                             3RD DIM: RECOMBINED METASTABLE INDEX
C OUTPUT : (R*8)  CCDA(,,,) = INTERPOLATION OF CCD COEFFICIENT (CM3 S-1)
```

```
C                              1ST DIM: TEMPERATURE INDEX
C                              2ND DIM: CHARGE STATE INDEX
C                              3RD DIM: RECOMBINING METASTABLE INDEX
C                              4TH DIM: RECOMBINED METASTABLE INDEX
C OUTPUT : (L*4)  LCCDA(,,) = .TRUE.  => CCD COEFFICIENT AVAILABLE
C                              .FALSE. => CCD COEFFICIENT NOT AVAILABLE
C                              1ST DIM: CHARGE STATE INDEX
C                              2ND DIM: RECOMBINING METASTABLE INDEX
C                              3RD DIM: RECOMBINED METASTABLE INDEX
C OUTPUT : (R*8)  PRBA(,,)  = INTERPOLATION OF PRB COEFFICIENT (W CM3 )
C                              1ST DIM: TEMPERATURE INDEX
C                              2ND DIM: CHARGE STATE INDEX
C                              3RD DIM: RECOMBINING METASTABLE INDEX
C OUTPUT : (L*4)  LCCDA(,)  = .TRUE.  => PRB COEFFICIENT AVAILABLE
C                              .FALSE. => PRB COEFFICIENT NOT AVAILABLE
C                              1ST DIM: CHARGE STATE INDEX
C                              2ND DIM: RECOMBINING METASTABLE INDEX
C OUTPUT : (R*8)  PRCA(,,)  = INTERPOLATION OF PRC COEFFICIENT (W CM3 )
C                              1ST DIM: TEMPERATURE INDEX
C                              2ND DIM: CHARGE STATE INDEX
C                              3RD DIM: RECOMBINING METASTABLE INDEX
C OUTPUT : (L*4)  LPRCA(,)  = .TRUE.  => PRC COEFFICIENT AVAILABLE
C                              .FALSE. => PRC COEFFICIENT NOT AVAILABLE
C                              1ST DIM: CHARGE STATE INDEX
C                              2ND DIM: RECOMBINING METASTABLE INDEX
C OUTPUT : (R*8)  QCDA(,,,) = INTERPOLATION OF QCD COEFFICIENT (CM3 S-1)
C                              1ST DIM: TEMPERATURE INDEX
C                              2ND DIM: CHARGE STATE INDEX
C                              3RD DIM: FIRST METASTABLE INDEX
C                              4TH DIM: SECOND METASTABLE INDEX
C OUTPUT : (L*4)  LQCDA(,,) = .TRUE.  => QCD COEFFICIENT AVAILABLE
C                              .FALSE. => QDC COEFFICIENT NOT AVAILABLE
C                              1ST DIM: CHARGE STATE INDEX
C                              2ND DIM: FIRST METASTABLE INDEX
C                              3RD DIM: SECOND METASTABLE INDEX
C OUTPUT : (R*8)  XCDA(,,,) = INTERPOLATION OF XCD COEFFICIENT (CM3 S-1)
C                              1ST DIM: TEMPERATURE INDEX
C                              2ND DIM: CHARGE STATE INDEX
C                              3RD DIM: FIRST PARENT METASTABLE INDEX
C                              4TH DIM: SECOND PARENT METASTABLE INDEX
C OUTPUT : (L*4)  LXCDA(,,) = .TRUE.  => XCD COEFFICIENT AVAILABLE
C                              .FALSE. => XDC COEFFICIENT NOT AVAILABLE
C                              1ST DIM: CHARGE STATE INDEX
C                              2ND DIM: FIRST PARENT METASTABLE INDEX
C                              3RD DIM: SECOND PARENT METASTABLE INDEX
C OUTPUT : (R*8)  PLTA(,,)  = INTERPOLATION OF PLT COEFFICIENT (W CM3 )
C                              1ST DIM: TEMPERATURE INDEX
C                              2ND DIM: CHARGE STATE INDEX
C                              3RD DIM: METASTABLE INDEX
C OUTPUT : (L*4)  LPLTA(,)  = .TRUE.  => PLT COEFFICIENT AVAILABLE
C                              .FALSE. => PLT COEFFICIENT NOT AVAILABLE
C                              1ST DIM: CHARGE STATE INDEX
C                              2ND DIM:  METASTABLE INDEX
C
C PROGRAM: (I*4)  IT        = GENERAL INDEX FOR TEMPERATURE
C          (I*4)  IZ        = GENERAL INDEX FOR CHARGE
C          (I*4)  IZ1       = GENERAL INDEX FOR CHARGE+1
C          (I*4)  IPRT      = GENERAL INDEX FOR PARENT METASTABLE
C          (I*4)  JPRT      = GENERAL INDEX FOR PARENT METASTABLE
C          (I*4)  IGRD      = GENERAL INDEX FOR METASTABLE
C          (I*4)  JGRD      = GENERAL INDEX FOR METASTABLE
```

```
C
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C
C
C AUTHOR : H. P. SUMMERS, JET
C          K1/1/57
C          JET  EXT. 4941
C
C DATE   : 25/04/94
C
C UPDATE :
C
C UNIX-IDL PORT:
C
C AUTHOR:  WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C
C DATE:    07/06/96
C
C VERSION: 1.1      DATE:07/06/96
C MODIFIED: WILLIAM OSBORN
C     - FIRST VERSION
C VERSION: 1.2      DATE:27/06/96
C MODIFIED: WILLIAM OSBORN
C     - REMOVED UNUSED VARIABLES
C VERSION: 1.3      DATE:27/06/96
C MODIFIED: WILLIAM OSBORN
C     - CHANGED 'DO  2 IT = 1,ITDIMD'
C           TO  'DO  2 IT = 1,NTDIM'
C SO AS NOT TO GO OUTSIDE ARRAY BOUNDS
C     - CHANGED DECISIONS TO 'IF ((LEXSA(I).OR.LDEFA(I)).AND.LSELA(I))'
C       SEE D5DATA FOR MORE INFO.
C
C-----------------------------------------------------------------------
      CHARACTER*120       DSFLLA(8)
      INTEGER             IPDIMD,      ISDIMD,      ITDIMD,      ITMAX
      INTEGER             IZ0,         IZ1MAX,      IZ1MIN,      IZDIMD
      INTEGER             NPART(IZDIMD),            NPARTR(IZDIMD)
      INTEGER             NTDIM
      LOGICAL             LACDA(IZDIMD,IPDIMD,IPDIMD)
      LOGICAL             LCCDA(IZDIMD,IPDIMD,IPDIMD)
      LOGICAL             LDEFA(8),    LEXSA(8),    LPART
      LOGICAL             LPLTA(IZDIMD,IPDIMD),     LPRBA(IZDIMD,IPDIMD)
      LOGICAL             LPRCA(IZDIMD,IPDIMD)
      LOGICAL             LQCDA(IZDIMD,IPDIMD,IPDIMD)
      LOGICAL             LSCDA(IZDIMD,IPDIMD,IPDIMD)
      LOGICAL             LSELA(8),    LXCDA(IZDIMD,IPDIMD,IPDIMD)
      REAL*8              ACDA(NTDIM,IZDIMD,IPDIMD,IPDIMD)
      REAL*8              CCDA(NTDIM,IZDIMD,IPDIMD,IPDIMD)
      REAL*8              DDENS(ITMAX),             DDENSD(ITDIMD)
      REAL*8              DRCOFD(ISDIMD,ITDIMD,ITDIMD)
      REAL*8              DRCOFI(ITMAX),            DTEV(ITMAX)
      REAL*8              DTEVD(ITDIMD)
      REAL*8              PLTA(NTDIM,IZDIMD,IPDIMD)
      REAL*8              PRBA(NTDIM,IZDIMD,IPDIMD)
      REAL*8              PRCA(NTDIM,IZDIMD,IPDIMD)
      REAL*8              QCDA(NTDIM,IZDIMD,IPDIMD,IPDIMD)
      REAL*8              SCDA(NTDIM,IZDIMD,IPDIMD,IPDIMD)
      REAL*8              XCDA(NTDIM,IZDIMD,IPDIMD,IPDIMD)
```

```
      REAL*8              ZDATA(ISDIMD)
```

5.30 d6mfll: Subroutine d6mfll from library adas4xx

```
       SUBROUTINE D6MFLL ( NTDIM  , IZDIMD , IPDIMD , IMDIMD ,
     &                     NMDIM  ,
     &                     NSTAGE , NPART  ,
     &                     ACDA   , SCDA   , CCDA   , QCDA   , XCDA   ,
     &                     DENS   , DENSH  ,
     &                     ITEM   ,
     &                     A
     &                   )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN 77 SUBROUTINE: D6MFLL *******************
C
C  PURPOSE: FILLS MATRIX WITH RECOMBINATION, IONISATION  AND METASTABLE
C           CROSS-COUPLING COEFFICIENTS READY FOR EIGENVECTOR SOLUTION
C
C  CALLING PROGRAM: D6MPOP
C
C  SUBROUTINE:
C
C
C INPUT  : (I*4)  NTDIM     = MAXIMUM NUMBER OF DTEV/DDENS PAIRS
C INPUT  : (I*4)  IZDIMD    = MAXIMUM NUMBER OF CHARGE STATES
C                             IN ISONUCLEAR MASTER FILES
C INPUT  : (I*4)  IPDIMD    = MAXIMUM NUMBER OF METASTABLES FOR EACH
C                             IONISATION STAGE
C INPUT  : (I*4)  IMDIMD    = MAXIMUM NUMBER OF POPULATIONS
C
C INPUT  : (I*4)  NMDIM     = MAX. NUMBER OF POPULATIONS (FROM D6MPOP)
C
C INPUT  : (I*4)  NSTAGE    = NUMBER OF IONISATION STATES (EXCL.
C                             EXTRA ONE (BARE NUCLEUS) ADDED AT END
C INPUT  : (I*4)  NPART()   = METASTABLE PARTITION.  I.E. NUMBER OF
C                             METASTABLES FROM CHARGE STATE IZ1MIN-1 TO
C                             IZ1MAX ON INPUT
C
C INPUT  : (R*8)  ACDA(,,,) = INTERPOLATION OF ACD COEFFICIENT (CM3 S-1)
C                             1ST DIM: TEMPERATURE INDEX
C                             2ND DIM: CHARGE STATE INDEX
C                             3RD DIM: RECOMBINING METASTABLE INDEX
C                             4TH DIM: RECOMBINED METASTABLE INDEX
C INPUT  : (R*8)  SCDA(,,,) = INTERPOLATION OF SCD COEFFICIENT (CM3 S-1)
C                             1ST DIM: TEMPERATURE INDEX
C                             2ND DIM: CHARGE STATE INDEX
C                             3RD DIM: RECOMBINING METASTABLE INDEX
C                             4TH DIM: RECOMBINED METASTABLE INDEX
C INPUT  : (R*8)  CCDA(,,,) = INTERPOLATION OF CCD COEFFICIENT (CM3 S-1)
C                             1ST DIM: TEMPERATURE INDEX
C                             2ND DIM: CHARGE STATE INDEX
C                             3RD DIM: RECOMBINING METASTABLE INDEX
C                             4TH DIM: RECOMBINED METASTABLE INDEX
C INPUT  : (R*8)  QCDA(,,,) = INTERPOLATION OF QCD COEFFICIENT (CM3 S-1)
C                             1ST DIM: TEMPERATURE INDEX
C                             2ND DIM: CHARGE STATE INDEX
C                             3RD DIM: FIRST METASTABLE INDEX
C                             4TH DIM: SECOND METASTABLE INDEX
C INPUT  : (R*8)  XCDA(,,,) = INTERPOLATION OF XCD COEFFICIENT (CM3 S-1)
C                             1ST DIM: TEMPERATURE INDEX
C                             2ND DIM: CHARGE STATE INDEX
C                             3RD DIM: FIRST PARENT METASTABLE INDEX
```

```
C                              4TH DIM: SECOND PARENT METASTABLE INDEX
C
C INPUT  : (R*8)  DENS()    = ELECTRON DENSITIES FOR MODEL
C INPUT  : (R*8)  DENSH()   = NEUTRAL HYDROGEN DENSITIES FOR MODEL
C
C INPUT  : (I*4)  ITEM      = CURRENT TEMP/DENSITY INDEX
C
C OUTPUT : (R*8)  A(,)      = RECOMB/IONIS COLL. RAD. MATRIX
C
C          (I*4)  I         = GENERAL INDEX
C          (I*4)  IGRD      = GENERAL INDEX
C          (I*4)  IND       = GENERAL INDEX
C          (I*4)  IND1      = GENERAL INDEX
C          (I*4)  IPRT      = GENERAL INDEX
C          (I*4)  IZ        = IONISATION STAGE COUNTER
C          (I*4)  I         = GENERAL INDEX
C          (I*4)  JGRD      = GENERAL INDEX
C
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C
C
C AUTHOR : H. P. SUMMERS, JET
C          K1/1/57
C          JET  EXT. 4941
C
C DATE   : 27/07/94
C UNIX-IDL PORT:
C
C AUTHOR:  WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C
C DATE:    07/06/96
C
C VERSION: 1.1      DATE:07/06/96
C MODIFIED: WILLIAM OSBORN
C     - FIRST VERSION
C
C VERSION: 1.2      DATE:28/03/07
C MODIFIED: ALLAN WHITEFORD
C     - REMOVED VERSIONING INFORMATION FROM COLUMNS 73-80
C
C-----------------------------------------------------------------------
      INTEGER             IMDIMD,      IPDIMD,      ITEM,        IZDIMD
      INTEGER             NMDIM,       NPART(IZDIMD),            NSTAGE
      INTEGER             NTDIM
      REAL*8              A(NMDIM,NMDIM)
      REAL*8              ACDA(NTDIM,IZDIMD,IPDIMD,IPDIMD)
      REAL*8              CCDA(NTDIM,IZDIMD,IPDIMD,IPDIMD)
      REAL*8              DENS(NTDIM), DENSH(NTDIM)
      REAL*8              QCDA(NTDIM,IZDIMD,IPDIMD,IPDIMD)
      REAL*8              SCDA(NTDIM,IZDIMD,IPDIMD,IPDIMD)
      REAL*8              XCDA(NTDIM,IZDIMD,IPDIMD,IPDIMD)
```

5.31 d6mpop: Subroutine d6mpop from library adas4xx

```
       SUBROUTINE D6MPOP( NTDIM  , IZDIMD , IPDIMD , IMDIMD ,
     &                    NSTAGE , ITMAX  , NPRT   , NMSUM  ,
     &                    ACDA   , SCDA   , CCDA   , QCDA   , XCDA   ,
     &                    DENS   , DENSH  ,
     &                    FABUN0 ,
     &                    ITEM   , TIMEF  ,
     &                    POPE   , POPF   , PINTE  , PINTF
     &                  )
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN 77 SUBROUTINE: D6MPOP *******************
C
C  PURPOSE: CALCULATION OF METASTABLE RESOLVED IONISATION STAGE
C           POPULATIONS OF A PARTICULAR ELEMENT FOR A GIVEN TEMPERATURE
C           AND DENSITY
C
C  CALLING PROGRAM: ADAS405
C
C  SUBROUTINE:
C
C
C INPUT :(I*4) NTDIM         = MAXIMUM NUMBER OF MODEL TEMPS/DENSITIES
C INPUT :(I*4) IZDIMD        = MAXIMUM NUMBER OF STAGES-1
C INPUT :(I*4) IPDIMD        = MAXIMUM SIZE OF METASTABLES FRO A STAGE
C INPUT :(I*4) IMDIMD        = MAXIMUM NUMBER OF POPULATIONS
C INPUT :(I*4) NSTAGE        = NUMBER OF STAGES-1
C INPUT :(I*4) ITMAX         = NUMBER OF MODEL TEMPS/DENSITIES
C INPUT :(R*8) NPRT( )       = PARTITION OF TOTAL METASTABLES ACCORDING
C                              TO IONISATION STAGES
C                              1ST DIM: STAGE INDEX
C INPUT :(I*4) NMSUM         = TOTAL NUMBER OF POPULATIONS
C INPUT :(R*8) ACDA(,,,)     = GENERALISED CR RECOMBINATION COEFFICIENT
C                              1ST DIM: TEMPERATURE INDEX
C                              2ND DIM: STAGE INDEX (LESS 1)
C                              3RD DIM: METASTABLE INDEX
C                              4TH DIM: METASTABLE INDEX
C INPUT :(R*8) SCDA(,,,)     = GENERALISED CR IONISATION COEFFICIENT
C                              1ST DIM: TEMPERATURE INDEX
C                              2ND DIM: STAGE INDEX (LESS 1)
C                              3RD DIM: METASTABLE INDEX
C                              4TH DIM: METASTABLE INDEX
C INPUT :(R*8) CCDA(,,,)     = GENERALISED CR CHARGE EXCH. COEFFICIENT
C                              1ST DIM: TEMPERATURE INDEX
C                              2ND DIM: STAGE INDEX (LESS 1)
C                              3RD DIM: METASTABLE INDEX
C                              4TH DIM: METASTABLE INDEX
C INPUT :(R*8) QCDA(,,,)     = GENERALISED CR CROSS-COUPL. COEFFICIENT
C                              1ST DIM: TEMPERATURE INDEX
C                              2ND DIM: STAGE INDEX (LESS 1)
C                              3RD DIM: METASTABLE INDEX
C                              4TH DIM: METASTABLE INDEX
C INPUT :(R*8) XCDA(,,,)     = GENERALISED CR PARENT X-CP. COEFFICIENT
C                              1ST DIM: TEMPERATURE INDEX
C                              2ND DIM: STAGE INDEX (LESS 1)
C                              3RD DIM: METASTABLE INDEX
C                              4TH DIM: METASTABLE INDEX
C INPUT :(R*8) DENS()        = ELECTRON DENSITIES FOR MODEL
C                              1ST DIM: TEMPERATURE INDEX
```

```
C INPUT :(R*8) DENSH()       = NEUTRAL HYDROGEN DENSITIES FOR MODEL
C                              1ST DIM: TEMPERATURE INDEX
C
C INPUT :(R*8) FABUN0()      = INITIAL POPULATION ABUNDANCES AT TIME = 0
C                              1ST DIM: POPULATION INDEX
C
C INPUT :(I*4) ITEM          = CURRENT TEMP/DENSITY INDEX
C INPUT :(R*8) TIMEF         = INTEGRATION TIME (SEC)
C
C OUTPUT:(R*8) POPE()        = IONISATION BALANCE POPULATIONS
C OUTPUT:(R*8) POPF()        = POPULATIONS AT T = TIMEF
C OUTPUT:(R*8) PINTE()       = POPULATION EXCESS INTEGRAL TO T = INFIN.
C OUTPUT:(R*8) PINTF()       = POPULATION EXCESS INTEGRAL TO T = TIMEF
C
C        (I*4) ISTATE        = STAGE INDEX
C        (I*4) ITEM          = GENERAL INDEX
C        (I*4) I             = GENERAL INDEX
C        (I*4) J             = GENERAL INDEX
C        (I*4) K             = GENERAL INDEX
C        (R*8) FV1()         = WORK ARRAY USED BY XXEIGN
C        (I*4) IV1()         = WORK ARRAY USED BY XXEIGN
C
C  ROUTINES:
C           ROUTINE    SOURCE    BRIEF DESCRIPTION
C           ------------------------------------------------------
C           D6MFLL     ADAS      FILL UP IONIS./RECOM. MATRIX
C           XXEIGN     ADAS/NETLIB GENERAL MATRIX DIAGONALISATION
C           XXSIM      ADAS/NETLIB SIMULTANEOUS EQUATION SOLUTION
C
C
C  AUTHOR:  H. P. SUMMERS, JET
C           K1/1/57
C           JET EXT. 4941
C
C  DATE:    27/06/94
C
C UNIX-IDL PORT:
C
C AUTHOR:  WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C
C DATE:    07/06/96
C
C VERSION: 1.1      DATE:07/06/96
C MODIFIED: WILLIAM OSBORN
C     - FIRST VERSION
C VERSION: 1.2      DATE:27/06/96
C MODIFIED: WILLIAM OSBORN
C     - REMOVED UNUSED VARIABLES
C VERSION: 1.3      DATE:13/09/96
C MODIFIED: WILLIAM OSBORN
C     - CHANGED PARAMETERS OF XXSIM IN LINE WITH CHANGES FOR
C             NON-SQUARE MATRICES.
C
C VERSION: 1.4 DATE: 13/05/2004
C MODIFIED: Martin O'Mullane
C - Increased NMDIM to 83 to accommodate lead.
C
C-----------------------------------------------------------------------
      INTEGER             IMDIMD,      IPDIMD,      ITEM,        ITMAX
      INTEGER             IZDIMD,      NMSUM,       NPRT(IZDIMD)
      INTEGER             NSTAGE,      NTDIM
```

```
REAL*8            ACDA(NTDIM,IZDIMD,IPDIMD,IPDIMD)
REAL*8            CCDA(NTDIM,IZDIMD,IPDIMD,IPDIMD)
REAL*8            DENS(NTDIM), DENSH(NTDIM)
REAL*8            FABUN0(IMDIMD),           PINTE(IMDIMD)
REAL*8            PINTF(IMDIMD),            POPE(IMDIMD)
REAL*8            POPF(IMDIMD)
REAL*8            QCDA(NTDIM,IZDIMD,IPDIMD,IPDIMD)
REAL*8            SCDA(NTDIM,IZDIMD,IPDIMD,IPDIMD),      TIMEF
REAL*8            XCDA(NTDIM,IZDIMD,IPDIMD,IPDIMD)
```

5.32 d6scrp: Subroutine d6scrp from library adas4xx

```
       SUBROUTINE D6SCRP( LRSCRP , LSNULL  ,
     &                    DSNINC , DSPECA  ,
     &                    NDLINE , NDCOMP  , NDRAT  , NDFILE ,
     &                    NFILE  , LFILE   ,
     &                    UID    , GROUP   , TYPE   , EXT    , ION    ,
     &                    MEMB   , IZ0     ,
     &                    NLINE  , NCOMP   ,
     &                    IZION  , IMET    , CIMET  , INDPH  , CINDPH ,
     &                    IFILE  , TITL    ,
     &                    NRAT   ,
     &                    ILINE  , JLINE   , TITR   , IRCODE
     &                  )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: D6SCRP  *********************
C
C  PURPOSE: TO READ SCRIPT FILE AND ACCESS EMISSIVITY DATA
C           ON SPECTRAL LINES REQUESTED FOR FURTHER PROCESSING IN
C           EQUILIBRIUM IONISATION CODES.
C
C  CALLING PROGRAM: ADAS406
C
C  SUBROUTINE:
C
C  INPUT : (C*80)  DSNINC   = SCRIPT DATA SET NAME (FULL MVS DSN)
C                             (IN FORM SUITABLE FOR DYNAMIC ALLOCATION)
C  INPUT : (I*4)   NDLINE   = MAXIMUM NUMBER OF LINES ALLOWED
C  INPUT : (I*4)   NDCOMP   = MAXIMUM NUMBER OF COMPONENT FOR EACH LINE
C  INPUT : (I*4)   NDRAT    = MAXIMUM NUMBER OF LINE RATIOS ALLOWED
C  INPUT : (I*4)   NDFILE   = MAXIMUM NUMBER OF EMISSIVITY FILES WHICH
C                             CAN BE SEARCHED
C
C  OUTPUT: (L*4)   LRSCRP   = .TRUE.  => SCRIPT FILE READ
C                             .FALSE. => SCRIPT FILE NOT READ
C  OUTPUT: (L*4)   LSNULL   = .TRUE.  => SCRIPT FILE SET TO NULL
C                             .FALSE. => SCRIPT FILE VALID
C  OUTPUT: (C*120) DSPECA() = PHOTON EMISSIVITY SOURCE FILES
C  OUTPUT: (I*4)   NFILE    = NUMBER OF PEC FILES TO BE SCANNED
C  OUTPUT: (L*4)   LFILE()  = .TRUE.  => PEC FILE EXISTS AND MATCHES
C                             .FALSE. => PEC FILE DOES NOT EXIST/MATCH
C  OUTPUT: (C*6)   UID()    = USER IDENTIFIER OF PEC FILE
C  OUTPUT: (C*8)   GROUP()  = GROUP IDENTIFIER OF PEC FILE
C  OUTPUT: (C*5)   TYPE()   = TYPE IDENTIFIER OF PEC FILE
C  OUTPUT: (C*3)   EXT()    = EXTENSION OF PEC FILE MEMBER NAME
C  OUTPUT: (C*4)   ION()    = ION NAME OF PEC FILE MEMBER NAME
C  OUTPUT: (C*8)   MEMB()   = MEMBER NAME OF PEC FILE
C  OUTPUT: (I*4)   NLINE    = NUMBER OF LINES IDENTIFIED IN SCRIPT
C  OUTPUT: (I*4)   NCOMP()  = NUMBER OF COMPONENTS OF SCRIPT LINE
C                             1ST DIM: LINE INDEX
C  OUTPUT: (I*4)   IZION(,) = CHARGE STATE OF COMPONENT
C                             1ST DIM: LINE INDEX
C                             2ND DIM: COMPONENT INDEX
C  OUTPUT: (I*4)   IMET(,)  = NUMBER OF COMPONENTS OF SCRIPT LINE
C                             1ST DIM: LINE INDEX
C                             2ND DIM: COMPONENT INDEX
C  OUTPUT: (C*1)   CIMET(,) = SIGN (+, BLANK OR -) OF METASTABLE
C                             1ST DIM: LINE INDEX
C                             2ND DIM: COMPONENT INDEX
C  OUTPUT: (I*4)   INDPH(,) = PEC FILE INDEX OF LINE COMPONENT
```

```
C                               1ST DIM: LINE INDEX
C                               2ND DIM: COMPONENT INDEX
C  OUTPUT: (C*1)   CINDPH(,)= DRIVER (E OR BLANK => ELECTRONS)
C                                     (H          => HYDROGEN )
C                               1ST DIM: LINE INDEX
C                               2ND DIM: COMPONENT INDEX
C  OUTPUT: (I*4)   IFILE(,) = INDEX OF PEC FILE IN FILE LIST
C                               1ST DIM: LINE INDEX
C                               2ND DIM: COMPONENT INDEX
C  OUTPUT: (C*12)  TITL(,)  = TITLE FOR LINE COMPONENT
C                               1ST DIM: LINE INDEX
C                               2ND DIM: COMPONENT INDEX
C  OUTPUT: (I*4)   NRAT     = NUMBER OF RATIOS IDENTIFIED IN SCRIPT
C  OUTPUT: (I*4)   ILINE()  = INDEX OF NUMERATOR LINE FOR LINE RATIO
C  OUTPUT: (I*4)   JLINE()  = INDEX OF DENOMINATOR LINE FOR LINE RATIO
C  OUTPUT: (C*25)  TITR()   = TILE FOR LINE RATIO
C  OUTPUT: (I*4)   IRCODE   = ERROR FLAG:
C                               0 => SCRIPT FILE WAS READ OKAY
C                               1 => SCRIPT FILE DOES NOT EXIST
C                               2 => I/O ERROR READING THE SCRIPT FILE
C                               3 => 1 OR MORE FILE NAMES IN SCRIPT FILE
C                                    IS/ARE INVALID.
C
C          (I*4)   IUNT10  = PARAMETER = INPUT UNIT FOR DATA
C          (L*4)   OPEN10  = .TRUE.  => FILE ALLOCATED TO UNIT 10.
C                            .FALSE. => NO FILE ALLOCATED TO UNIT 10.
C
C ROUTINES:
C          ROUTINE    SOURCE   BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          XXSLEN     ADAS      FIND NON-BLANK CHARACTERS IN STRING
C
C
C AUTHOR:  H. P. SUMMERS, JET
C          K1/1/57
C          JET EXT. 4941
C
C DATE:    20/04/94
C
C UNIX-IDL PORT:
C
C AUTHOR:  WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C
C DATE:    07/06/96
C
C VERSION: 1.1      DATE:07/06/96
C MODIFIED: WILLIAM OSBORN
C     - FIRST VERSION. USED SOME CODE FROM D5SCRP.FOR V1.3 IN
C             ADDING IRCODE PARAMETER.
C VERSION: 1.2      DATE:27/06/96
C MODIFIED: WILLIAM OSBORN
C     - REMOVED UNUSED VARIABLES
C VERSION: 1.3      DATE:27/06/96
C MODIFIED: WILLIAM OSBORN
C     - INCREASED LENGTH OF CLINE AND DSNPEC TO 120
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      CHARACTER           CIMET(NDLINE,NDCOMP)
      CHARACTER           CINDPH(NDLINE,NDCOMP)
```

```
CHARACTER*80        DSNINC
CHARACTER*120       DSPECA(NDFILE)
CHARACTER*3         EXT(NDFILE)
CHARACTER*8         GROUP(NDFILE)
CHARACTER*4         ION(NDFILE)
CHARACTER*8         MEMB(NDFILE)
CHARACTER*12        TITL(NDLINE,NDCOMP)
CHARACTER*25        TITR(NDRAT)
CHARACTER*5         TYPE(NDFILE)
CHARACTER*6         UID(NDFILE)
INTEGER             IFILE(NDLINE,NDCOMP),     ILINE(NDRAT)
INTEGER             IMET(NDLINE,NDCOMP),      INDPH(NDLINE,NDCOMP)
INTEGER             IRCODE,      IZ0,         IZION(NDLINE,NDCOMP)
INTEGER             JLINE(NDRAT),             NCOMP(NDLINE)
INTEGER             NDCOMP,      NDFILE,      NDLINE,      NDRAT
INTEGER             NFILE,       NLINE,       NRAT
LOGICAL             LFILE(NDFILE),            LRSCRP,      LSNULL
```

5.33 d6sgcf: Subroutine d6sgcf from library adas4xx

```
       SUBROUTINE D6SGCF( IZ0    , IZL    , IZH    ,
     &                    ISDIMD , IZDIMD , ITDIMD , IPDIMD , IMDIMD ,
     &                    NMSUM  , IZIP   , IMIP   , IPIZM  ,
     &                    NDLINE , NDCOMP ,
     &                    NLINE  , NCOMP  , SPECL  , IPLINE ,
     &                    IZION  , IMET   , CIMET  , INDPH  , CINDPH ,
     &                    IFILE    ,
     &                    NTDIM  , ITMAX  ,
     &                    DENS   , DENSH  ,
     &                    PECA   , LPEC   ,
     &                    FPABUN ,
     &                    GCFPEQ , GCFEQ  ,
     &                    NDRAT  , NRAT   ,
     &                    ILINE  , JLINE  ,
     &                    RATA
     &                  )
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: D6SGCF *********************
C
C PURPOSE : TO ASSEMBLE GCF FUNCTIONS AND THEIR COMPONENTS USING
C           FRACTIONAL METASTABLE ABUNDANCES.
C
C
C
C INPUT  : (I*4)  IZ0       = NUCLEAR CHARGE
C INPUT  : (I*4)  IZL       = MINIMUM ION CHARGE+1 IN MASTER DATA FILES
C INPUT  : (I*4)  IZH       = MAXIMUM ION CHARGE+1 IN MASTER DATA FILES
C INPUT  : (I*4)  ISDIMD    = MAXIMUM NUMBER OF (CHARGE, PARENT, GROUND)
C                             BLOCKS IN ISONUCLEAR MASTER FILES
C INPUT  : (I*4)  IZDIMD    = MAXIMUM NUMBER OF CHARGE STATES
C                             IN ISONUCLEAR MASTER FILES
C INPUT  : (I*4)  ITDIMD    = MAXIMUM NUMBER OF TEMP OR DENS VALUES IN
C                             ISOELECTRONIC MASTER FILES
C INPUT  : (I*4)  IPDIMD    = MAXIMUM NUMBER OF METASTABLES FOR EACH
C                             IONISATION STAGE
C INPUT  : (I*4)  IMDIMD    = MAXIMUM NUMBER OF METASTABLES
C
C INPUT  : (I*4)  NMSUM     = TOTAL NUMBER OF POPULATIONS
C
C INPUT  :        IZIP()    = ION CHARGE +1 (IZ1) OF METASTABLE IN LIST
C INPUT  :        IMIP()    = METASTABLE INDEX WITHIN CHARGE STATE IZ1
C                             OF METASTABLE INDEX FROM COMPLETE LIST
C INPUT  :        IPIZM(,)  = METASTABLE INDEX IN COMPLETE LIST
C                             1ST DIM: INDEX IZ1-IZL+1
C                             2ND DIM: METASTABLE COUNT FOR STAGE (IGRD)
C INPUT  : (I*4)  NDLINE    = MAXIMUM NUMBER OF LINES ALLOWED
C INPUT  : (I*4)  NDCOMP    = MAXIMUM NUMBER OF COMPONENT FOR EACH LINE
C INPUT  : (I*4)  NLINE     = NUMBER OF LINES IDENTIFIED IN SCRIPT
C INPUT  : (I*4)  NCOMP()   = NUMBER OF COMPONENTS OF SCRIPT LINE
C INPUT  : (I*4)  IZION(,)  = CHARGE STATE OF COMPONENT
C                             1ST DIM: LINE INDEX
C                             2ND DIM: COMPONENT INDEX
C INPUT  : (I*4)  IMET(,)   = METASTABLE INDEX OF COMPONENT OF
C                             SCRIPT LINE WITHIN CHARGE STATE
C                             1ST DIM: LINE INDEX
C                             2ND DIM: COMPONENT INDEX
C INPUT  : (C*1)  CIMET(,)  = SIGN (+, BLANK OR -) OF METASTABLE
```

```
C                               1ST DIM: LINE INDEX
C                               2ND DIM: COMPONENT INDEX
C INPUT  : (I*4)  INDPH(,)  = PEC FILE INDEX OF LINE COMPONENT
C                               1ST DIM: LINE INDEX
C                               2ND DIM: COMPONENT INDEX
C INPUT  : (C*1   CINDPH(,) = DRIVER (E OR BLANK => ELECTRONS)
C                                     (H          => HYDROGEN )
C                                1ST DIM: LINE INDEX
C                                2ND DIM: COMPONENT INDEX
C INPUT  : (I*4)  IFILE(,)  = INDEX OF PEC FILE IN FILE LIST
C                               1ST DIM: LINE INDEX
C                               2ND DIM: COMPONENT INDEX
C INPUT  : (I*4)  NTDIM     = MAXIMUM NUMBER OF DTEV/DDENS PAIRS
C INPUT  : (I*4)  ITMAX     = NUMBER OF ( DTEV() , DDENS() ) PAIRS
C INPUT  : (R*8)  DENS()    = ELECTRON DENSITIES (CM-3))
C INPUT  : (R*8)  DENSH()   = HYDROGEN DENSITIES (CM-3))
C INPUT  : (R*8)  PECA(,,)  = PHOTON EMISSIVITY COEFFICIENTS (CM3 S-1)
C                               1ST DIM: TEMPERATURE INDEX
C                               2ND DIM: LINE INDEX
C                               3RD DIM: COMPONENT INDEX
C INPUT  : (L*4)  LPEC(,)   = .TRUE.  => PHOTON EMISSIVITY OBTAINED
C                               .FALSE. => PHOTON EMISSIVITY NOT OBTAINED
C                               2ND DIM: LINE INDEX
C                               3RD DIM: COMPONENT INDEX
C INPUT  : (R*8)  FPABUN(,) = RESOLVED METASTABLE EQUILIBRIUM
C                               FRACTIONAL ABUNDANCES
C                               1ST DIM: - TEMPERATURE/DENSITY PAIR
C                               2ND DIM: - METASTABLE INDEX
C INPUT  : (I*4)  NDRAT     = MAXIMUM NUMBER OF LINE RATIOS ALLOWED
C INPUT  : (I*4)  NRAT      = NUMBER OF RATIOS IDENTIFIED IN SCRIPT
C INPUT  : (I*4)  ILINE()   = INDEX OF NUMERATOR LINE FOR LINE RATIO
C INPUT  : (I*4)  JLINE()   = INDEX OF DENOMINATOR LINE FOR LINE RATIO
C
C OUTPUT : (C*16) SPECL(,)  = SPEC. OF POINTERS OF LINE COMPONENT
C                               1ST DIM: LINE INDEX
C                               2ND DIM: COMPONENT INDEX
C OUTPUT : (I*4)  IPLINE(,) = METASTABLE POINTER OF LINE COMPONENT
C                               1ST DIM: LINE INDEX
C                               2ND DIM: COMPONENT INDEX
C OUTPUT : (R*8)  GCFPEQ(,,)= GCF FUNC. COMPONENT (CM3 S-1)
C                               1ST DIM: TEMPERATURE INDEX
C                               2ND DIM: LINE INDEX
C                               3ND DIM: LINE COMPONENT INDEX
C OUTPUT : (R*8)  GCFEQ(,)  = GCF FUNCTION  (CM3 S-1)
C                               1ST DIM: TEMPERATURE INDEX
C                               2ND DIM: LINE INDEX
C OUTPUT : (R*8)  RATA(,)   = LINE GCF RATIOS
C                               1ST IND: TEMPERATURE INDEX
C                               2ND IND: RATIO INDEX
C
C
C PROGRAM: (I*4)  IT        = GENERAL INDEX FOR TEMPERATURE
C          (I*4)  IP        = GENERAL INDEX FOR CHARGE
C          (I*4)  IZ1       = GENERAL INDEX FOR CHARGE+1
C          (I*4)  IL        = GENERAL INDEX FOR LINE
C          (I*4)  IR        = GENERAL INDEX FOR RATIO
C          (I*4)  ICPT      = GENERAL INDEX FOR LINE COMPONENT
C
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
```

```
C          ------------------------------------------------------------
C
C
C AUTHOR : H. P. SUMMERS, JET
C          K1/1/57
C          JET  EXT. 4941
C
C DATE   : 03/05/94
C
C UNIX-IDL PORT:
C
C AUTHOR:  WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C
C DATE:    07/06/96
C
C VERSION: 1.1      DATE:07/06/96
C MODIFIED: WILLIAM OSBORN
C     - FIRST VERSION
C VERSION: 1.2      DATE:27/06/96
C MODIFIED: WILLIAM OSBORN
C     - REMOVED UNUSED VARIABLES
C
C-----------------------------------------------------------------------
      CHARACTER           CIMET(NDLINE,NDCOMP)
      CHARACTER           CINDPH(NDLINE,NDCOMP)
      CHARACTER*16        SPECL(NDLINE,NDCOMP)
      INTEGER             IFILE(NDLINE,NDCOMP),     ILINE(NDRAT)
      INTEGER             IMDIMD,       IMET(NDLINE,NDCOMP)
      INTEGER             IMIP(IMDIMD),             INDPH(NDLINE,NDCOMP)
      INTEGER             IPDIMD,       IPIZM(IZDIMD,IPDIMD)
      INTEGER             IPLINE(NDLINE,NDCOMP),    ISDIMD,       ITDIMD
      INTEGER             ITMAX,        IZ0,        IZDIMD,       IZH
      INTEGER             IZION(NDLINE,NDCOMP),     IZIP(IMDIMD)
      INTEGER             IZL,          JLINE(NDRAT)
      INTEGER             NCOMP(NDLINE),            NDCOMP,       NDLINE
      INTEGER             NDRAT,        NLINE,      NMSUM,        NRAT
      INTEGER             NTDIM
      LOGICAL             LPEC(NDLINE,NDCOMP)
      REAL*8              DENS(NTDIM), DENSH(NTDIM)
      REAL*8              FPABUN(NTDIM,IMDIMD),     GCFEQ(NTDIM,NDLINE)
      REAL*8              GCFPEQ(NTDIM,NDLINE,NDCOMP)
      REAL*8              PECA(NTDIM,NDLINE,NDCOMP)
      REAL*8              RATA(NTDIM,NDRAT)
```

5.34 d6spec: Subroutine d6spec from library adas4xx

```
       SUBROUTINE D6SPEC( LRSPEC ,
     &                    NDLINE , NDCOMP  , NDRAT  , NDFILE ,
     &                    NFILE  , LFILE   ,
     &                    UID    , GROUP   , TYPE   , EXT    ,
     &                    IZ0    , DSPECA  ,
     &                    NLINE  , NCOMP   ,
     &                    IZION  , IMET    , CIMET  , INDPH  ,
     &                    IFILE  ,
     &                    NTDIM  , ITMAX  ,
     &                    TEIN   , DEIN    , THIN   , DHIN   ,
     &                    PECA   ,
     &                    LPEC   , LTRNG   , LDRNG
     &                  )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: D6SPEC  *******************
C
C  PURPOSE: TO CALCULATE PHOTON EMISSIVITY COEFFICIENTS FOR
C           SPECTRAL LINES IDENTIFIED IN SCRIPT FILE
C
C  CALLING PROGRAM: ADAS406
C
C  SUBROUTINE:
C
C  INPUT : (I*4)   NDLINE   = MAXIMUM NUMBER OF LINES ALLOWED
C  INPUT : (I*4)   NDCOMP   = MAXIMUM NUMBER OF COMPONENT FOR EACH LINE
C  INPUT : (I*4)   NDRAT    = MAXIMUM NUMBER OF LINE RATIOS ALLOWED
C  INPUT : (I*4)   NDFILE   = MAXIMUM NUMBER OF EMISSIVITY FILES WHICH
C                             CAN BE SEARCHED
C  INPUT : (I*4)   NFILE    = NUMBER OF PEC FILES TO BE SCANNED
C  INPUT : (L*4)   LFILE()  = .TRUE.  => PEC FILE EXISTS AND MATCHES
C                             .FALSE. => PEC FILE DOES NOT EXIST/MATCH
C  INPUT : (C*6)   UID()    = USER IDENTIFIER OF PEC FILE
C  INPUT : (C*8)   GROUP()  = GROUP IDENTIFIER OF PEC FILE
C  INPUT : (C*5)   TYPE()   = TYPE IDENTIFIER OF PEC FILE
C  INPUT : (C*3)   EXT()    = EXTENSION OF PEC FILE MEMBER NAME
C  INPUT : (I*4)   IZ0      = NUCLEAR CHARGE OF IMPURITY
C  INPUT : (C*120) DSPECA() = PHOTON EMISSIVITY SOURCE FILES
C  INPUT : (I*4)   NLINE    = NUMBER OF LINES IDENTIFIED IN SCRIPT
C  INPUT : (I*4)   NCOMP()  = NUMBER OF COMPONENTS OF SCRIPT LINE
C                             1ST DIM: LINE INDEX
C  INPUT : (I*4)   IZION(,) = CHARGE STATE OF COMPONENT
C                             1ST DIM: LINE INDEX
C                             2ND DIM: COMPONENT INDEX
C  INPUT : (I*4)   IMET(,)  = NUMBER OF COMPONENTS OF SCRIPT LINE
C                             1ST DIM: LINE INDEX
C                             2ND DIM: COMPONENT INDEX
C  INPUT : (C*1)   CIMET(,) = SIGN (+, BLANK OR -) OF METASTABLE
C                             1ST DIM: LINE INDEX
C                             2ND DIM: COMPONENT INDEX
C  INPUT : (I*4)   INDPH(,) = PEC FILE INDEX OF LINE COMPONENT
C                             1ST DIM: LINE INDEX
C                             2ND DIM: COMPONENT INDEX
C  INPUT : (I*4)   IFILE(,) = INDEX OF PEC FILE IN FILE LIST
C                             1ST DIM: LINE INDEX
C                             2ND DIM: COMPONENT INDEX
C  INPUT : (I*4)   NTDIM    = MAXIMUM NUMBER OF TEMP/DENSITY SETS
C  INPUT : (I*4)   ITMAX    = NUMBER OF TEMP/DENSITY SETS
C  INPUT : (R*8)   TEIN()   = ELECTRON TEMPERATURES (EV)
```

```
C  INPUT : (R*8)   DEIN()   = ELECTRON DENSITIES (CM-3)
C  INPUT : (R*8)   THIN()   = HYDROGEN TEMPERATURES (EV)
C  INPUT : (R*8)   DHIN()   = HYDROGEN DENSITIES (CM-3)
C
C  OUTPUT: (L*4)   LRSPEC   = .TRUE.  => PEC PROCESSING DONE
C                             .FALSE. => PEC PROCESSING NOT DONE
C  OUTPUT: (R*8)   PECA(,,) = PHOTON EMISSIVITY COEFFICIENTS (CM3 S-1)
C                             1ST DIM: TEMPERATURE INDEX
C                             2ND DIM: LINE INDEX
C                             3RD DIM: COMPONENT INDEX
C  OUTPUT: (L*4)   LPEC(,)  = .TRUE.  => PHOTON EMISSIVITY OBTAINED
C                             .FALSE. => PHOTON EMISSIVITY NOT OBTAINED
C                             2ND DIM: LINE INDEX
C                             3RD DIM: COMPONENT INDEX
C
C          (I*4)   IUNT10  = PARAMETER = INPUT UNIT FOR DATA
C          (L*4)   OPEN10  = .TRUE.  => FILE ALLOCATED TO UNIT 10.
C                            .FALSE. => NO FILE ALLOCATED TO UNIT 10.
C
C ROUTINES:
C          ROUTINE    SOURCE   BRIEF DESCRIPTION
C          ------------------------------------------------------------
C    D5SPC2   IDL-ADAS  OBTAIN PHOTON EMISSIVITY COEFFICIENT
C
C AUTHOR:  H. P. SUMMERS, JET
C          K1/1/57
C          JET EXT. 4941
C
C DATE:    20/04/94
C
C UNIX-IDL PORT:
C
C AUTHOR:  WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C
C DATE:    07/06/96
C
C VERSION: 1.1      DATE:07/06/96
C MODIFIED: WILLIAM OSBORN
C     - FIRST VERSION
C VERSION: 1.2      DATE:27/06/96
C MODIFIED: WILLIAM OSBORN
C     - REMOVED UNUSED VARIABLES
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      CHARACTER           CIMET(NDLINE,NDCOMP)
      CHARACTER*120       DSPECA(NDFILE)
      CHARACTER*3         EXT(NDFILE)
      CHARACTER*8         GROUP(NDFILE)
      CHARACTER*5         TYPE(NDFILE)
      CHARACTER*6         UID(NDFILE)
      INTEGER             IFILE(NDLINE,NDCOMP),     IMET(NDLINE,NDCOMP)
      INTEGER             INDPH(NDLINE,NDCOMP),     ITMAX,       IZ0
      INTEGER             IZION(NDLINE,NDCOMP),     NCOMP(NDLINE)
      INTEGER             NDCOMP,      NDFILE,      NDLINE,      NDRAT
      INTEGER             NFILE,       NLINE,       NTDIM
      LOGICAL             LDRNG(NTDIM),             LFILE(NDFILE)
      LOGICAL             LPEC(NDLINE,NDCOMP),      LRSPEC
      LOGICAL             LTRNG(NTDIM)
      REAL*8              DEIN(NTDIM), DHIN(NTDIM)
```

```
REAL*8              PECA(NTDIM,NDLINE,NDCOMP)
REAL*8              TEIN(NTDIM), THIN(NTDIM)
```

5.35 d6spow: Subroutine d6spow from library adas4xx

```
       SUBROUTINE D6SPOW( LSELA  , LEXSA  , LDEFA  , LPART  , LEXSS  ,
     &                    IZ0    , IZL    , IZH    , NPART  ,
     &                    ISDIMD , IZDIMD , ITDIMD , IPDIMD , IMDIMD ,
     &                    ACDA   , SCDA   , CCDA   , PRBA   ,
     &                    PRCA   , QCDA   , XCDA   , PLTA   ,
     &                    NMSUM  , IZIP   , IMIP   , IPIZM  ,
     &                    NTDIM  , ITMAX  ,
     &                    DENS   , DENSH  ,
     &                    FPABUN , FSABUN , FPINTG , FSINTG ,
     &                    ELTPEQ ,
     &                    ACDSEQ , SCDSEQ , CCDSEQ , ERBSEQ ,
     &                    ERCSEQ , ELTSEQ ,
     &                    ERBEQ  , ERCEQ  , ELTEQ  , ERADA
     &                  )
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: D6SPOW *********************
C
C PURPOSE : TO ASSEMBLE RADIATED ENERGY EXCESS FUNCTIONS USING
C           FRACTIONAL METASTABLE ABUNDANCES INTEGRAL EXCESSES
C
C NOTE    : THE SOURCE ISONUCLEAR MASTER FILE DATA ARE OBTAINED BY A
C           PRIOR CALL TO SUBROUTINE D6DATA FROM SEQUENTIAL FILES
C           WITH THE FOLLOWING NAMING CONVENTIONS:
C
C                   (1) JETSHP.ACD<YR>#<EL).<CODE>DATA
C                   (2) JETSHP.SCD<YR>#<EL>.<CODE>DATA
C                   (3) JETSHP.CCD<YR>#<EL>.<CODE>DATA
C                   (4) JETSHP.PRB<YR>#<EL>.<FILT>.<CODE>DATA
C                   (5) JETSHP.PRC<YR>#<EL>.<FILT>.<CODE>DATA
C                   (6) JETSHP.QCD<YR>#<EL>.<CODE>DATA
C                   (7) JETSHP.XCD<YR>#<EL>.<CODE>DATA
C                   (8) JETSHP.PLT<YR>#<EL>.<CODE>DATA
C
C       WHERE, <YR>   = TWO DIGIT YEAR NUMBER
C              <EL>   = ONE OR TWO CHARACTER ELEMENT SYMBOL
C              <CODE> = R       => PARTIAL DATA
C                       U       => PARTIAL DATA
C                       OMITTED => STANDARD DATA
C              <FILT> = SIX CHARACTER POWER FILTER CODE
C
C       AND DATA OF CLASSES 6 AND 7 DO NOT EXIST FOR THE PARTIAL CASE.
C
C
C INPUT  : (L*4)  LSELA()   = .TRUE.  => INPUT DATA SET TYPE FOR THIS
C                                        INDEX SELECTED
C                           = .FALSE. => INPUT DATA SET FOR THIS INDEX
C                                        NOT SELECTED
C INPUT  : (L*4)  LEXSA()   = .TRUE.  => INPUT DATA SET TYPE FOR THIS
C                                        SELECTED INDEX EXISTS
C                           = .FALSE. => INPUT DATA SET DOES NOT EXIST
C                                        FOR THIS SELECTED INDEX
C INPUT  : (L*4)  LDEFA()   = .TRUE.  => INPUT DATA SET TYPE FOR THIS
C                                        DEFAULT YEAR INDEX EXISTS
C                           = .FALSE. => INPUT DATA SET DOES NOT EXIST
C                                        FOR THIS DEFAULT YEAR INDEX
C INPUT  : (L*4)  LPART     = .TRUE.  => PARTIAL DATA SELECTED
C                           = .FALSE. => STANDARD DATA SELECTED
```

```
C INPUT  : (I*4)  IZ0       = NUCLEAR CHARGE
C INPUT  : (I*4)  IZL       = MINIMUM ION CHARGE+1 IN MASTER DATA FILES
C INPUT  : (I*4)  IZH       = MAXIMUM ION CHARGE+1 IN MASTER DATA FILES
C INPUT  : (I*4)  NPART()   = METASTABLE PARTITION.  I.E. NUMBER OF
C                             METASTABLES FROM CHARGE STATE IZL-1 TO
C                             IZH ON INPUT
C INPUT  : (I*4)  ISDIMD    = MAXIMUM NUMBER OF (CHARGE, PARENT, GROUND)
C                             BLOCKS IN ISONUCLEAR MASTER FILES
C INPUT  : (I*4)  IZDIMD    = MAXIMUM NUMBER OF CHARGE STATES
C                             IN ISONUCLEAR MASTER FILES
C INPUT  : (I*4)  ITDIMD    = MAXIMUM NUMBER OF TEMP OR DENS VALUES IN
C                             ISOELECTRONIC MASTER FILES
C INPUT  : (I*4)  IPDIMD    = MAXIMUM NUMBER OF METASTABLES FOR EACH
C                             IONISATION STAGE
C INPUT  : (I*4)  IMDIMD    = MAXIMUM NUMBER OF METASTABLES
C
C INPUT  : (R*8)  ACDA(,,,) = INTERPOLATION OF ACD COEFFICIENT (CM3 S-1)
C                             1ST DIM: TEMPERATURE INDEX
C                             2ND DIM: CHARGE STATE INDEX
C                             3RD DIM: RECOMBINING METASTABLE INDEX
C                             4TH DIM: RECOMBINED METASTABLE INDEX
C INPUT  : (R*8)  SCDA(,,,) = INTERPOLATION OF SCD COEFFICIENT (CM3 S-1)
C                             1ST DIM: TEMPERATURE INDEX
C                             2ND DIM: CHARGE STATE INDEX
C                             3RD DIM: RECOMBINING METASTABLE INDEX
C                             4TH DIM: RECOMBINED METASTABLE INDEX
C INPUT  : (R*8)  CCDA(,,,) = INTERPOLATION OF CCD COEFFICIENT (CM3 S-1)
C                             1ST DIM: TEMPERATURE INDEX
C                             2ND DIM: CHARGE STATE INDEX
C                             3RD DIM: RECOMBINING METASTABLE INDEX
C                             4TH DIM: RECOMBINED METASTABLE INDEX
C INPUT  : (R*8)  PRBA(,,)  = INTERPOLATION OF PRB COEFFICIENT (W CM3 )
C                             1ST DIM: TEMPERATURE INDEX
C                             2ND DIM: CHARGE STATE INDEX
C                             3RD DIM: RECOMBINING METASTABLE INDEX
C INPUT  : (R*8)  PRCA(,,)  = INTERPOLATION OF PRC COEFFICIENT (W CM3 )
C                             1ST DIM: TEMPERATURE INDEX
C                             2ND DIM: CHARGE STATE INDEX
C                             3RD DIM: RECOMBINING METASTABLE INDEX
C INPUTT : (R*8)  QCDA(,,,) = INTERPOLATION OF QCD COEFFICIENT (CM3 S-1)
C                             1ST DIM: TEMPERATURE INDEX
C                             2ND DIM: CHARGE STATE INDEX
C                             3RD DIM: FIRST METASTABLE INDEX
C                             4TH DIM: SECOND METASTABLE INDEX
C INPUT  : (R*8)  XCDA(,,,) = INTERPOLATION OF XCD COEFFICIENT (CM3 S-1)
C                             1ST DIM: TEMPERATURE INDEX
C                             2ND DIM: CHARGE STATE INDEX
C                             3RD DIM: FIRST PARENT METASTABLE INDEX
C                             4TH DIM: SECOND PARENT METASTABLE INDEX
C INPUT  : (R*8)  PLTA(,,)  = INTERPOLATION OF PLT COEFFICIENT (W CM3 )
C                             1ST DIM: TEMPERATURE INDEX
C                             2ND DIM: CHARGE STATE INDEX
C                             3RD DIM: METASTABLE INDEX
C INPUT  : (I*4)  NMSUM     = TOTAL NUMBER OF POPULATIONS
C
C INPUT  :        IZIP()    = ION CHARGE +1 (IZ1) OF METASTABLE IN LIST
C INPUT  :        IMIP()    = METASTABLE INDEX WITHIN CHARGE STATE IZ1
C                             OF METASTABLE INDEX FROM COMPLETE LIST
C INPUT  :        IPIZM(,)  = METASTABLE INDEX IN COMPLETE LIST
C                             1ST DIM: INDEX IZ1-IZL+1
C                             2ND DIM: METASTABLE COUNT FOR STAGE (IGRD)
```

```
C INPUT  : (I*4)  NTDIM     = MAXIMUM NUMBER OF DTEV/DDENS PAIRS
C INPUT  : (I*4)  ITMAX     = NUMBER OF ( DTEV() , DDENS() ) PAIRS
C INPUT  : (R*8)  DENS()    = ELECTRON DENSITIES (CM-3))
C INPUT  : (R*8)  DENSH()   = HYDROGEN DENSITIES (CM-3))
C INPUT  : (R*8)  FPABUN(,) = RESOLVED METASTABLE EQUILIBRIUM
C                             FRACTIONAL ABUNDANCES
C                             1ST DIM: - TEMPERATURE/DENSITY PAIR
C                             2ND DIM: - METASTABLE INDEX
C INPUT  : (R*8)  FPINTG(,) = RESOLVED TRANSIENT METASTABLE POPULATION
C                             EXCESS INTEGRALS
C                             1ST DIM: - TEMPERATURE/DENSITY PAIR
C                             2ND DIM: - METASTABLE INDEX
C OUTPUT : (L*4)  LEXSS()   = .TRUE.  => OUTPUT STANDARD MASTER DATA FOR
C                                        THIS INDEX GENERATED
C                           = .FALSE. => OUTPUT STANDARD MASTER DATA FOR
C                                        THIS INDEX NOT GENERATED
C OUTPUT : (R*8)  FSABUN(,) = STAGE EQUILIBRIUM FRACTIONAL ABUNDANCES
C                             1ST DIM: - TEMPERATURE/DENSITY PAIR INDEX
C                             2ND DIM: - CHARGE STATE INDEX (IZ1-IZL+1)
C OUTPUT : (R*8)  FSINTG(,) = STAGE TRANSIENT FRACTIONAL ABUNDANCES
C                             EXCESSES
C                             1ST DIM: - TEMPERATURE/DENSITY PAIR INDEX
C                             2ND DIM: - CHARGE STATE INDEX (IZ1-IZL+1)
C OUTPUT : (R*8)  ELTPEQ(,) = METASTABLE PARTIAL TRANSIENT RADIATED
C                             LINE ENERGY EXCESS FUNCTIONS
C                             1ST DIM: - TEMPERATURE/DENSITY PAIR
C                             2ND DIM: - METASTABLE INDEX
C OUTPUT : (R*8)  ACDSEQ(,) = STANDARD (UNRESOLVED) ACD COEFFICIENT
C                             1ST DIM: - TEMPERATURE/DENSITY PAIR INDEX
C                             2ND DIM: - CHARGE STATE INDEX (IZ1-IZL+1)
C OUTPUT : (R*8)  SCDSEQ(,) = STANDARD (UNRESOLVED) SCD COEFFICIENT
C                             1ST DIM: - TEMPERATURE/DENSITY PAIR INDEX
C                             2ND DIM: - CHARGE STATE INDEX (IZ1-IZL+1)
C OUTPUT : (R*8)  CCDSEQ(,) = STANDARD (UNRESOLVED) CCD COEFFICIENT
C                             1ST DIM: - TEMPERATURE/DENSITY PAIR INDEX
C                             2ND DIM: - CHARGE STATE INDEX (IZ1-IZL+1)
C OUTPUT : (R*8)  ERBSEQ(,) = STANDARD (UNRESOLVED) RB ENERGY EXCESS
C                             COEFFICIENT
C                             1ST DIM: - TEMPERATURE/DENSITY PAIR INDEX
C                             2ND DIM: - CHARGE STATE INDEX (IZ1-IZL+1)
C OUTPUT : (R*8)  ERCSEQ(,) = STANDARD (UNRESOLVED) RC ENERGY EXCESS
C                             COEFFICIENT
C                             1ST DIM: - TEMPERATURE/DENSITY PAIR INDEX
C                             2ND DIM: - CHARGE STATE INDEX (IZ1-IZL+1)
C OUTPUT : (R*8)  ELTSEQ(,) = STANDARD (UNRESOLVED) LT ENERGY EXCESS
C                             COEFFICIENT
C                             1ST DIM: - TEMPERATURE/DENSITY PAIR INDEX
C                             2ND DIM: - CHARGE STATE INDEX (IZ1-IZL+1)
C OUTPUT : (R*8)  ERBEQ()   = TOTAL TRANSIENT RADIATED RECOM-BREMS
C                                   ENERGY EXCESS FUNCTION
C OUTPUT : (R*8)  ERCEQ()   = TOTAL TRANSIENT CX RADIATED RECOM ENERGY
C                                   EXCESS FUNCTION NORMALISED TO
C                                   ELECTRON DENSITY
C OUTPUT : (R*8)  ELTEQ()   = TOTAL TRANSIENT RADIATED LINE ENERGY
C                                   EXCESS FUNCTION
C OUTPUT : (R*8)  ERADA()   = TOTAL TRANSIENT RADIATED ENERGY EXCESS
C                                   FUNCTION
C
C PROGRAM: (I*4)  IT        = GENERAL INDEX FOR TEMPERATURE
C          (I*4)  IP        = GENERAL INDEX FOR CHARGE
C          (I*4)  IZ1       = GENERAL INDEX FOR CHARGE+1
```

```
C          (I*4)  ICL       = GENERAL INDEX FOR CLASS
C          (I*4)  IPP       = GENERAL PARENT INDEX
C          (I*4)  IPG       = GENERAL GROUND INDEX
C          (I*4)  IZREF     = GENERAL CHARGE STAE POINTER INDEX
C          (I*4)  IPRT      = GENERAL INDEX FOR PARENT METASTABLE
C          (I*4)  IGRD      = GENERAL INDEX FOR METASTABLE
C
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C
C
C AUTHOR : H. P. SUMMERS, JET
C          K1/1/57
C          JET  EXT. 4941
C
C DATE   : 28/04/94
C
C UNIX-IDL PORT:
C
C AUTHOR:  WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C
C DATE:    07/06/96
C
C VERSION: 1.1      DATE:07/06/96
C MODIFIED: WILLIAM OSBORN
C     - FIRST VERSION
C VERSION: 1.2      DATE:27/06/96
C MODIFIED: WILLIAM OSBORN
C     - REMOVED UNUSED VARIABLES
C
C-----------------------------------------------------------------------
      INTEGER             IMDIMD,      IMIP(IMDIMD),             IPDIMD
      INTEGER             IPIZM(IZDIMD,IPDIMD),     ISDIMD,      ITDIMD
      INTEGER             ITMAX,       IZ0,         IZDIMD,      IZH
      INTEGER             IZIP(IMDIMD),             IZL,         NMSUM
      INTEGER             NPART(IZDIMD),            NTDIM
      LOGICAL             LDEFA(8),    LEXSA(8),    LEXSS(8),    LPART
      LOGICAL             LSELA(8)
      REAL*8              ACDA(NTDIM,IZDIMD,IPDIMD,IPDIMD)
      REAL*8              ACDSEQ(NTDIM,IZDIMD)
      REAL*8              CCDA(NTDIM,IZDIMD,IPDIMD,IPDIMD)
      REAL*8              CCDSEQ(NTDIM,IZDIMD),     DENS(NTDIM)
      REAL*8              DENSH(NTDIM),             ELTEQ(NTDIM)
      REAL*8              ELTPEQ(NTDIM,IMDIMD),     ELTSEQ(NTDIM,IZDIMD)
      REAL*8              ERADA(NTDIM),             ERBEQ(NTDIM)
      REAL*8              ERBSEQ(NTDIM,IZDIMD),     ERCEQ(NTDIM)
      REAL*8              ERCSEQ(NTDIM,IZDIMD),     FPABUN(NTDIM,IMDIMD)
      REAL*8              FPINTG(NTDIM,IZDIMD),     FSABUN(NTDIM,IZDIMD)
      REAL*8              FSINTG(NTDIM,IZDIMD)
      REAL*8              PLTA(NTDIM,IZDIMD,IPDIMD)
      REAL*8              PRBA(NTDIM,IZDIMD,IPDIMD)
      REAL*8              PRCA(NTDIM,IZDIMD,IPDIMD)
      REAL*8              QCDA(NTDIM,IZDIMD,IPDIMD,IPDIMD)
      REAL*8              SCDA(NTDIM,IZDIMD,IPDIMD,IPDIMD)
      REAL*8              SCDSEQ(NTDIM,IZDIMD)
      REAL*8              XCDA(NTDIM,IZDIMD,IPDIMD,IPDIMD)
```

5.36 d7alfs: Subroutine d7alfs from library adas4xx

```
      SUBROUTINE D7ALFS( NDMET  , NDCONF , NDTHET , ndlev  , ndtrn  ,
     &                   IODIMD , IPMET  , IPMETR , IZ     , IZ0    ,
     &                   NTHETA , THETA  , ITREF  , IGRPA  ,
     &                   NORB   , VORB   , iz1    ,
     &                   il     , ia     , isa    , xja    , wa     ,
     &                   cstrga , icnte  , aval   , ie1a   , ie2a   ,
     &                   NCUT   , N0A    , PARMR  ,
     &                   ALFRA  , ALFRA0 , ALFRAR , ALRAPX ,
     &                   NCONFG ,
     &                   WVMIN  , WVMAX  ,
     &                   ECF    , FCF    , PCF    , WCF    , W      ,
     &                   NCF    , LCF    , NDCF   , LDCF   , NDMIN  ,
     &                   E      , DE0    , DE     , FM0    , FM     ,
     &                   IINAA  , IIPNAA , NCTAA  , NCTAAC , ECTAA  ,
     &                   NTRANS , ITYPE  , N1A    , NCUTT  , PARMD  ,
     &                   ALFDA  , ALFPART, AGNGPX ,
     &                   KGRPA  , NSYS
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: D7ALFS *********************
C
C
C  PURPOSE:
C           (1) CACLULATES RADIATIVE AND DIELECTRONIC VALUES AND
C               PARAMETERS FROM SPECIFIC ION FILES WHICH HAVE EISSNER
C               CONFIGURATION NOTATION
C           (2) CONSIDERS METASTABLE LEVEL INDICES AND EVALUATES NCUT
C               IDENTIFIES DIPOLE TRANSITION OF TYPE DN=0 AND DN>0
C               EVALUATES OSCILLATOR STRENGTHS AND AVERAGE ENERGY
C               OF TRANSITION
C           (3) SEPARATES TRANSITION ARRAYS FOR EACH METASTABLE AND
C               EVALUATES WAVELENGTH RANGES OF TRANSITION ARRAYS
C           (4) EVALUATES POWER IN EACH TRANSTION ARRAY
C           (5) CALCULATES IONISATION, RADIATIVE & DIELECTRONIC
C               RECOMBINATION VALUES AND PARAMETERS.
C
C
C  CALLING PROGRAM: ADAS407
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  IUNIT    = UNIT NUMBER FOR SPECIFIC ION FILE FOR
C  INPUT : (I*4)  NDMET    = MAXIMUM NUMBER OF METASTABLES ALLOWED
C  INPUT : (I*4)  NDTHET   = MAXIMUM NUMBER OF TEMPS. FOR MAINCL FILE
C  INPUT : (I*4)  NDCONF   = MAXIMUM NUMBER OF CONFIGURATIONS OR
C                            DIPOLE TRANSITIONS PER PARENT ALLOWED
C  INPUT : (I*4)  IODIMD   = MAXIMUM NUMBER OF ORBITALS
C
C  INPUT : (I*4)  NPMET    = NO. OF RECOMBINING ION (PARENT) METASTABLES
C  INPUT : (I*4)  IPMETR() = INDICES OF RECOMBINING ION (PARENT)
C                            METASTABLES IN LEVEL LIST
C
C  INPUT : (I*4)  IZ       = RECOMBINING ION CHARGE
C  INPUT : (I*4)  IZ0      = NUCLEAR CHARGE
C
C  INPUT : (I*4)  NTHETA   = NUMBER OF TEMPERATURES FOR MAINCL FILE
C  INPUT : (R*8)  THETA()  = Z-SCALED TEMPERATURES FOR MAINCL FILE
C  INPUT : (I*4)  ITREF    = MAINCL TEMPERATURE INDEX FOR MATCHING
```

```
C
C  INPUT : (I*4)  IGRPA()  = NUMBER OF ELECTRONS ALLOWED IN EACH SHELL
C                             1ST DIM: SHELL INDEX (1=1S, 2=2S ETC)
C
C  INPUT : (R*8)  NORB     = NUMBER OF ELECTRON ORBITALS REQUIRED
C  INPUT : (R*8)  VORB()   = EFFECT. PRINC. QUANT. NO. FOR ORBITAL
C                             1ST DIM: SHELL INDEX (1=1S, 2=2S ETC)
C  OUTPUT: (I*4)  N0A(,)   = LOWEST ALLOWED N-SHELL
C                             1ST DIM: PARENT INDEX
C                             2ND DIM: SPIN SYSTEM INDEX
C  I/O   : (R*4)  PARMR(,,)= PARAMETERS OF RADIATIVE RECOMBINATION
C                             APPROXIMATE FORMS
C                             1ST DIM: PARENT INDEX
C                             2ND DIM: SPIN SYSTEM INDEX
C                             3RD DIM: PARMS.  1: EFF. N FOR LOWEST LEVEL
C                                              2: PHASE SPACE FACTOR
C                                              3: ENERGY DISPLACEMENT
C                                              4: SCALING MULTIPLIER
C  OUTPUT: (R*8)  ALFRA(,,)= TOTAL RADIATIVE RECOMB. COEFFTS. (CM3 S-1)
C                             1ST DIM: PARENT INDEX
C                             2ND DIM: SPIN SYSTEM INDEX
C                             3RD DIM: TEMPERATURE INDEX
C  OUTPUT: (R*8)  ALFRA0(,,)=GROUND RADIATIVE RECOMB. COEFFTS. (CM3 S-1)
C                             1ST DIM: PARENT INDEX
C                             2ND DIM: SPIN SYSTEM INDEX
C                             3RD DIM: TEMPERATURE INDEX
C  OUTPUT: (R*8)  ALFRAR(,,)=EXCIT. RADIATIVE RECOMB. COEFFTS. (CM3 S-1)
C                             1ST DIM: PARENT INDEX
C                             2ND DIM: SPIN SYSTEM INDEX
C                             3RD DIM: TEMPERATURE INDEX
C  OUTPUT: (R*8)  ALFRAPX() =
C  OUTPUT: (I*4)  NCONFG   = NUMBER OF CONFIGURATIONS
C
C  OUTPUT: (R*8)  WVMIN(,) = MINIMUM WAVELENGTH FOR TRANSITION ARRAY (A)
C                             1ST DIM: PARENT INDEX
C                             2ND DIM: CONFIGURATION INDEX
C  OUTPUT: (R*8)  WVMAX(,) = MAXIMUM WAVELENGTH FOR TRANSITION ARRAY (A)
C                             1ST DIM: PARENT INDEX
C                             2ND DIM: CONFIGURATION INDEX
C  OUTPUT: (R*8)  ECF(,)   = AVERAGE ENERGY FOR TRANSITION ARRAY (RYD)
C                             1ST DIM: PARENT INDEX
C                             2ND DIM: CONFIGURATION INDEX
C  OUTPUT: (R*8)  FCF(,)   = SUMMED OSCIL. STRENGTH FOR TRANSITION ARRAY
C                             1ST DIM: PARENT INDEX
C                             2ND DIM: CONFIGURATION INDEX
C  OUTPUT: (R*8)  PCF(,)   = RADIATED POWER FOR TRANSITION ARRAY AT
C                             SELELECTED TEMPERATURE (CF. ITSEL)
C                             1ST DIM: PARENT INDEX
C                             2ND DIM: CONFIGURATION INDEX
C  OUTPUT: (R*8)  WCF(,)   = STATISTICAL WEIGHT FOR UPPER SHELL OF ARRAY
C                             1ST DIM: PARENT INDEX
C                             2ND DIM: CONFIGURATION INDEX
C  OUTPUT: (R*8)  W()      = STATISTICAL WEIGHT FOR PARENT
C                             1ST DIM: PARENT INDEX
C  OUTPUT: (I*4)  NCF(,)   = N-SHELL OF ACTIVE ELEC. IN PARENT FOR ARRAY
C                             1ST DIM: PARENT INDEX
C                             2ND DIM: CONFIGURATION INDEX
C  OUTPUT: (I*4)  LCF(,)   = L-SHELL OF ACTIVE ELEC. IN PARENT FOR ARRAY
C                             1ST DIM: PARENT INDEX
C                             2ND DIM: CONFIGURATION INDEX
C  OUTPUT: (I*4)  NDCF(,)  = N-SHELL CHANGE OF ACTIVE ELECTRON FOR ARRAY
```

```
C                               1ST DIM: PARENT INDEX
C                               2ND DIM: CONFIGURATION INDEX
C  OUTPUT: (I*4)  LDCF(,)   = L-SHELL CHANGE OF ACTIVE ELECTRON FOR ARRAY
C                               1ST DIM: PARENT INDEX
C                               2ND DIM: CONFIGURATION INDEX
C  OUTPUT: (I*4)  NDMIN()   = ?
C                               1ST DIM: PARENT INDEX
C  OUTPUT: (R*8)  E()       = AVERAGE ENERGY FOR TRANSITION ARRAY (RYD)
C                               1ST DIM: PARENT INDEX
C  OUTPUT: (R*8)  DE0()     = MEAN DELTA N = 0 TRANSITION ENERGY (CM-1)
C                               1ST DIM: PARENT INDEX
C  OUTPUT: (R*8)  DE()      = MEAN DELTA N > 0 TRANSITION ENERGY (CM-1)
C                               1ST DIM: PARENT INDEX
C  OUTPUT: (R*8)  FM0()     = DELTA N = 0 OSCILLATOR STRENGTH
C                               1ST DIM: PARENT INDEX
C  OUTPUT: (R*8)  FM()      = DELTA N > 0 OSCILLATOR STRENGTH
C                               1ST DIM: PARENT INDEX
C  OUTPUT: (I*4)  IINAA(,) = UPP. LEVEL INDEX OF DIPOLE TRANS FOR ARRAY
C                               1ST DIM: PARENT INDEX
C                               2ND DIM: CONFIGURATION INDEX
C  OUTPUT: (I*4)  IIPNAA(,)= PAR. LEVEL INDEX OF DIPOLE TRANS FOR ARRAY
C                               1ST DIM: PARENT INDEX
C                               2ND DIM: CONFIGURATION INDEX
C  OUTPUT: (I*4)  NCTAA(,) = ?
C                               1ST DIM: PARENT INDEX
C                               2ND DIM: CONFIGURATION INDEX
C  OUTPUT: (I*4)  NCTAAC(,)= SECONDARY AUGER N-SHELL CUT-OFF FOR ARRAY
C                               1ST DIM: PARENT INDEX
C                               2ND DIM: CONFIGURATION INDEX
C  OUTPUT: (R*8)  ECTAA (,)= IONIS. ENERGY CUT-OFF (CM-1) FOR ARRAY
C                               1ST DIM: PARENT INDEX
C                               2ND DIM: CONFIGURATION INDEX
C
C  OUTPUT: (I*4)  NTRANS() = NUMBER OF DIPOLE TRANSITIONS FOR PARENT
C                               1ST DIM: PARENT INDEX
C  OUTPUT: (I*4)  ITYPE(,) = TYPE OF DIELECTRONIC CORE TRANSITION
C                               1ST DIM: PARENT INDEX
C                               2ND DIM: CONFIGURATION INDEX
C  OUTPUT: (I*4)  N1A(,)   = LOWEST ALLOWED N-SHELL VIA DIELECTRONIC
C                               TRANSITION
C                               1ST DIM: PARENT INDEX
C                               2ND DIM: CONFIGURATION INDEX
C  OUTPUT: (I*4)  NCUTT()  = N-SHELL ALT. AUG. CUT-OFF FOR DIELECTRONIC
C                               TRANSITION
C                               1ST DIM: PARENT INDEX
C                               2ND DIM: CONFIGURATION INDEX
C  OUTPUT: (R*4)  PARMD(,,)= PARAMETERS OF DIELECTRONIC RECOMBINATION
C                               APPROXIMATE FORMS
C                               1ST DIM: PARENT INDEX
C                               3RD DIM: PARMS.  1: EFF. N FOR LOWEST LEVEL
C                                                2: PHASE SPACE FACTOR
C                                                3: ENERGY DISPLACEMENT
C                                                4: SCALING MULTIPLIER
C                                                5: EFF. N FOR LOWEST LEVEL
C                                                6: PHASE SPACE FACTOR
C                                                7: ENERGY DISPLACEMENT
C                                                8: SCALING MULTIPLIER
C                                                9: ENERGY DISPLACEMENT
C                                               10: SCALING MULTIPLIER
C                               3RD DIM: CONFIGURATION INDEX
C  OUTPUT: (R*8)  ALFDA(,) = TOTAL DIELECTRONIC RECOMB. COEFFTS. WITHOUT
```

```
C                             SPIN SYSTEM DIVISION  (CM3 S-1)
C                             1ST DIM: PARENT INDEX
C                             3RD DIM: TEMPERATURE INDEX
C  OUTPUT: (R*8)  ALFPART(,)=PARTIAL DIELECT. RECOMB. COEFFTS. (CM3 S-1)
C                             1ST DIM: PARENT INDEX
C                             2ND DIM: CONFIGURATION INDEX
C                             3RD DIM: TEMPERATURE INDEX
C  OUTPUT: (R*8)  AGNGPX() = ?
C
C  OUTPUT: (I*4)  NIA()    = ?
C  OUTPUT: (I*4)  LIA()    = ?
C  OUTPUT: (R*8)  WIA()    = ?
C  OUTPUT: (I*4)  NIA()    = ?
C  OUTPUT: (I*4)  LIA()    = ?
C  OUTPUT: (R*8)  WIA()    = ?
C  OUTPUT: (R*8)  WIA()    = ?
C
C  OUTPUT: (I*4)  KGRPA()  = INDEX POINTER TO ELECTRON ORBITALS.
C  OUTPUT: (I*4)  NSYS()   = NUMBER OF SPIN SYSTEMS (1 OR 2)
C                             1ST DIM: PARENT INDEX
C  OUTPUT: (I*4)  NCUT()   = N-SHELL CUT-OFF
C                             1ST DIM: PARENT INDEX
C
C
C NOTE:
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          NVGOEL     ADAS      CALC. RAD. RECOM. COEFFTS. TO N-SHELLS
C          GPCALL     ADAS      CALC. DIELECTRONIC COEFFTS. TO N-SHELLS
C
C AUTHOR:  H. P. SUMMERS, JET
C          K1/1/57
C          JET EXT. 4941
C
C DATE:    29/06/94
C
C UPDATES: 09/08/94 - HPS - CORRECT PARAMETER LIST FOR GPCALL TO INCLUDE
C                           IFSEL.  SET IFSEL =1, BUT IT IS NOT USED
C UPDATES: 07/03/96 - HPS - PUT DIMENSIONALITY OF A NUMBER OF VECTORS OF
C                           LENGTH 15 TO INTERNAL PARAMETER ITDIMD
C
C UPDATES: 15/03/96 - PEB - CORRECT ERROR: WIA(,) AND WJA(,) CHANGED TO
C                           REAL*8 FROM INTEGER.
C
C UNIX-IDL PORT:
C    WILLIAM OSBORN, TESSELLA SUPPORT SERVICES PLC.
C
C DATE:    22ND APRIL 1996
C
C VERSION: 1.1 DATE: 22-04-96
C MODIFIED: WILLIAM OSBORN
C     - FIRST VERSION. NO CHANGES TO IBM VERSION
C
C VERSION: 1.2 DATE: 14-05-96
C MODIFIED: WILLIAM OSBORN
C     REARRANGED ARGUMENTS TO STAY UNDER
C     LIMIT OF 20 CONTINUATION CHARACTERS AT ARCETRI AND GARCHING
C
C VERSION: 1.3 DATE: 20-08-96
```

```
C MODIFIED: HUGH SUMMERS + WILLIAM OSBORN
C      ADDED TRAP FOR ZERO POWER FOR A PARENT. THIS
C           IS NOT A FULL SOLUTION (CF. D7AUTS COMMENTS)
C VERSION: 1.4 DATE: 24-06-97
C MODIFIED: HUGH SUMMERS
C           CHANGED PARAMETER KTERM FROM 100 TO 300
C                             KTRAN FROM 1000 TO 3200
C
C
C VERSION : 1.5
C DATE    : 23-05-2003
C MODIFIED: Martin O'Mullane
C               - Pass in adf04 data rather than rewinding and
C                 reading it again.
C               - Remove all unused variables and reduced length of
C                 parameter list.
C               - Remove redundant code and format statements.
C
C
C VERSION : 1.6
C DATE    : 04-11-2003
C MODIFIED: Hugh Summers
C               - Increased extended electron list strings to 93
C               - Changed minimum A-value, aminsc to 1.0D3
C               - match iodimd to ndorb for internal dimensions
C
C VERSION : 1.6
C DATE    : 06-01-2004
C MODIFIED: Martin O'Mullane
C               - Remove redundant nia, lia, wia, nja, lja and wja
C                 arguments.
C               - Pre-process configuration string with a new
C                 routine (ceprep) to account for leading d10 and
C                 f10-f14 terms.
C               - Add error trapping code to check for overruns
C                 and index=0 errors.
C
C VERSION : 1.7
C DATE    : 15-11-2004
C MODIFIED: Martin O'Mullane
C               - Increase to 3500 levels and 500000 transitions.
C
C-----------------------------------------------------------------------
      CHARACTER*18        CSTRGA(NDLEV)
      INTEGER             IA(NDLEV),   ICNTE,       IE1A(NDTRN)
      INTEGER             IE2A(NDTRN), IGRPA(IODIMD)
      INTEGER             IINAA(NDMET,NDCONF),      IIPNAA(NDMET,NDCONF)
      INTEGER             IL,          IODIMD,      IPMET
      INTEGER             IPMETR(NDMET),            ISA(NDLEV),  ITREF
      INTEGER             ITYPE(NDMET,NDCONF),      IZ,          IZ0
      INTEGER             IZ1,         KGRPA(IODIMD)
      INTEGER             LCF(NDMET,NDCONF),        LDCF(NDMET,NDCONF)
      INTEGER             N0A(NDMET,2),             N1A(NDMET,NDCONF)
      INTEGER             NCF(NDMET,NDCONF),        NCONFG
      INTEGER             NCTAA(NDMET,NDCONF),      NCTAAC(NDMET,NDCONF)
      INTEGER             NCUT(NDMET), NCUTT(NDMET,NDCONF)
      INTEGER             NDCF(NDMET,NDCONF),       NDCONF,      NDLEV
      INTEGER             NDMET,       NDMIN(NDMET),            NDTHET
      INTEGER             NDTRN,       NORB,        NSYS(NDMET), NTHETA
      INTEGER             NTRANS(NDMET)
      REAL*8              AGNGPX(ITDIMD),           ALFDA(NDMET,NDTHET)
```

```
REAL*8              ALFPART(NDMET,NDCONF,NDTHET)
REAL*8              ALFRA(NDMET,2,NDTHET)
REAL*8              ALFRA0(NDMET,2,NDTHET)
REAL*8              ALFRAR(NDMET,2,NDTHET),    ALRAPX(ITDIMD)
REAL*8              AVAL(NDTRN), DE(NDMET),    DE0(NDMET)
REAL*8              E(NDMET),    ECF(NDMET,NDCONF)
REAL*8              ECTAA(NDMET,NDCONF),       FCF(NDMET,NDCONF)
REAL*8              FM(NDMET),   FM0(NDMET)
REAL*8              PARMD(NDMET,10,NDCONF),    PARMR(NDMET,2,4)
REAL*8              PCF(NDMET,NDCONF),         THETA(NDTHET)
REAL*8              VORB(IODIMD),              W(NDMET)
REAL*8              WA(NDLEV),   WCF(NDMET,NDCONF)
REAL*8              WVMAX(NDMET,NDCONF),       WVMIN(NDMET,NDCONF)
REAL*8              XJA(NDLEV)
```

5.37 d7auts: Subroutine d7auts from library adas4xx

```
      SUBROUTINE D7AUTS( NDMET  , NDTHET , NDORB  , ndlev  , ndqdn  ,
     &                   IODIMD ,
     &                   IZ     , IZ0    , iz1    ,
     &                   il     , ia     , isa    , ila    , xja    ,
     &                   wa     , cstrga , bwno   , iorb   , qdorb  ,
     &                   NTHETA , THETA  ,
     &                   NPMET  ,
     &                   IGRPA  , EICHR  , LEICHR ,
     &                   NSYS   , ISPSYS , INPAR  , ILPAR  , ENPAR  ,
     &                   SBCHA  ,
     &                   KGRPA  , IZETA4 , EIONA  , NZETA  ,
     &                   IONLEV , XITRUE ,
     &                   NORB   , VORB   ,
     &                   N0A    , PARMR  ,
     &                   LLINK  , ILINK  , LEISS  ,
     &                   NMET   , IMETR
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: D7AUTS *********************
C
C
C  PURPOSE:     CALCULATES IONISATION RATES FROM GIVEN SPECIFIC ION FILE
C               USING BURGESS/CHIDICHIMO FORMULA RESOLVED INTO
C               PARENT AND SPIN SYSTEM COMPONENTS.
C
C               AUTOIONISATION EFFECTS ARE INCLUDED BY REDUCING ORBITAL
C               IONISATION ENERGY TO EXCITATION ENERGY OF LOWEST AUTO-
C               IONISING STATE, WITH A LINEAR SWITCHOFF
C               BETWEEN 20<Z1<25.
C
C  CALLING PROGRAM: ADAS407
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  IUNIT    = UNIT NUMBER FOR SPECIFIC ION FILE FOR
C                            RECOMBINED ION
C  INPUT : (I*4)  NDMET    = MAXIMUM NUMBER OF METASTABLES ALLOWED
C  INPUT : (I*4)  NDTHET   = MAXIMUM NUMBER OF TEMPS. FOR MAINCL FILE
C  INPUT : (I*4)  NDORB    = MAXIMUM NUMBER OF ELECTRON ORBITALS
C  INPUT : (I*4)  IODIMD   = MAXIMUM NUMBER OF ELECTRON ORBITALS
C
C  INPUT : (I*4)  NTHETA   = NUMBER OF TEMPERATURES FOR MAINCL FILE
C  INPUT : (R*8)  THETA()  = Z-SCALED TEMPERATURES FOR MAINCL FILE
C
C  INPUT : (I*4)  NPMET    = NO. OF RECOMBINING ION (PARENT) METASTABLES
C
C  INPUT : (I*4)  NSYS()   = NUMBER OF SPIN SYSTEMS FOR RECOMBINED ION
C                            1ST DIM: PARENT INDEX
C  INPUT : (I*4)  ISPSYS(,)= RECOMBINED ION SPIN
C                            1ST DIM: PARENT INDEX
C                            2ND IND: SPIN SYSTEM INDEX
C  INPUT : (I*4)  INPAR()  = N QUANTUM NO. SUM FOR ELECTRONS OF PARENT
C                            1ST DIM: PARENT INDEX
C  INPUT : (I*4)  ILPAR()  = L QUANTUM NO. SUM FOR ELECTRONS OF PARENT
C                            1ST DIM: PARENT INDEX
C  INPUT : (R*8)  ENPAR()  = RECOMBINING ION (PARENT) ENERGY
C                            1ST DIM: PARENT INDEX
```

```
C  INPUT : (L*4)  LLINK(,,)= .TRUE.  => LINK EXISTS
C                            .FALSE. => NO LINK EXISTS
C                             1ST DIM: METASTABLE INDEX
C                             2ND DIM: PARENT METASTABLE INDEX
C                             3RD DIM: SPEN SYSTEM INDEX
C  INPUT : (L*4)  ILINK(,,)= DECIMAL ORBITAL INDEX FOR RECOMBINED
C                            ION ORBITAL DIFFERENCE WITH PARENT
C                             1ST DIM: METASTABLE INDEX
C                             2ND DIM: PARENT METASTABLE INDEX
C                             3RD DIM: SPEN SYSTEM INDEX
C  INPUT : (L*4)  LEISS    = .TRUE. => PARENTS AND METASTABLES FOUND
C                                      TO HAVE EISSNER CONFIG. FORMS
C                            .FALSE => NOT EISSNER CONFIG. FORMS
C  INPUT : (I*4)  NMET    = NUMBER OF METASTABLES (1 <= NMET <= 'NDMET')
C  INPUT : (I*4)  IMETR() = INDEX OF METASTABLE IN COMPLETE LEVEL LIST
C                           (ARRAY SIZE = 'NDMET' )
C
C
C  OUTPUT: (I*4)  IZ       = CHARGE ON IONISING ION
C  OUTPUT: (I*4)  IZ0      = NUCLEAR CHARGE
C
C  OUTPUT: (I*4)  NORB     = NUMBER OF DISTINCT ELECTRON ORBITALS FROM
C                            CONFIGURATIONS IN TERM LIST
C  OUTPUT: (R*8)  VORB()   = EFFECTIVE PRINCIPAL QUANTUM NUMBERS OF
C                            ORBITALS (ORDERED)
C  OUTPUT: (R*8)  EPSIL()  = ENERGIES OF ORBITALS (RYDBERG) (NATURAL
C                            ORDER)
C  OUTPUT: (I*4)  KGRPA()  = INDEXING OF SORTED ORBITALS TO NATURAL
C                            ORDER
C  OUTPUT: (I*4)  N0A(,)   = LOWEST ALLOWED N-SHELL
C                            1ST DIM: PARENT INDEX
C                            2ND IND: SPIN SYSTEM INDEX
C  OUTPUT: (R*8)  PARMR(,,)= PARAMETERS OF RADIATIVE RECOMBINATION
C                            APPROXIMATE FORMS
C                            1ST DIM: PARENT INDEX
C                            2ND IND: SPIN SYSTEM INDEX
C                            3RD IND: PARMS.  1: EFF. N FOR LOWEST LEVEL
C                                             2: PHASE SPACE FACTOR
C                                             3: ENERGY DISPLACEMENT
C                                             4: SCALING MULTIPLIER
C
C  OUTPUT: (R*8)  SBCHA(,,)= BEST EXTIMATE OF METASTABLE AVERAGED
C                            IONISATION RATE
C                            1ST DIM: PARENT INDEX
C                            2ND IND: SPIN SYSTEM INDEX
C                            3RD IND: TEMPERATURE INDEX
C  OUTPUT  (R*8)  C1       = SCALING FACTOR
C  OUTPUT  (I*4)  IZETA4(,,)= NO. OF ELECTRONS IN ORBITALS
C                            1ST DIM: PARENT INDEX
C                            2ND IND: SPIN SYSTEM INDEX
C                            3RD IND: SHELL INDEX
C  OUTPUT  (R*8)  EIONA(,,)= SET OF SCALED ORBITAL IONISATION POTENTIALS
C                            1ST DIM: PARENT INDEX
C                            2ND IND: SPIN SYSTEM INDEX
C                            3RD IND: SHELL INDEX
C  OUTPUT  (I*4)  NZETA(,) = NO. OF OCCUPIED GROUND STATE ORBITALS
C                            1ST DIM: PARENT INDEX
C                            2ND IND: SPIN SYSTEM INDEX
C  OUTPUT  (I*4)  IONLEV(,)= TERM INDEX OF GROUND FOR SPIN SYSTEM/PARENT
C                            1ST DIM: PARENT INDEX
C                            2ND IND: SPIN SYSTEM INDEX
```

```
C  OUTPUT  (I*4)  XITRUE(,)= EXACT IONISATION ENERGY OF GROUND FOR SPIN
C                           SYSTEM/PARENT
C                           1ST DIM: PARENT INDEX
C                           2ND IND: SPIN SYSTEM INDEX
C
C          (I*4)  KTERM    = PARAMETER = MAXIMUM NUMBER OF TERMS
C                                        ALLOWED IN SUBROUTINE
C          (I*4)  NTRUE    = NUMBER OF ORBITALS USED IN LEVEL LIST
C                           INCLUDING NON-DISPLAYED CLOSE SHELLS
C
C NOTE:
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          SBCHID     ADAS      CALCULATES IONISATION RATES
C          I4JGRP     ADAS      DECIMAL VALUE OF EISSNER HEX CHARACTER
C          I4LGRP     ADAS      L-VALUE FROM EISSNER HEX ORBITAL
C          I4NGRP     ADAS      N-VALUE FROM EISSNER HEX ORBITAL
C          I4NDEC     ADAS      N-VALUE FROM DECIMAL ORBITAL
C          I4UNIT     ADAS      FETCH UNIT NUMBER FOR MESSAGE OUTPUT
C
C AUTHOR:  H. P. SUMMERS, JET
C          K1/1/57
C          JET EXT. 4941
C
C DATE:    29/06/94
C
C UNIX-IDL PORT:
C    WILLIAM OSBORN, TESSELLA SUPPORT SERVICES PLC.
C
C DATE:    22ND APRIL 1996
C
C VERSION: 1.1 DATE: 22-04-96
C MODIFIED: WILLIAM OSBORN
C     - FIRST VERSION. NO CHANGES TO IBM CODE.
C
C VERSION: 1.2 DATE: 20-08-96
C MODIFIED: HUGH SUMMERS + WILLIAM OSBORN
C     - TRAP IONLEV(IPAR,ISYS)=0. NOTE NO
C             FULL SOLUTION YET TO THIS PROBLEM OF
C             PARENTS AND RECMBINED METASTABLES WITH
C             INNER SHELL DIFFERENCES
C           - ADDED FOLLOWING TO CALL PARAMETERS
C             LLINK,ILINK,LEISS,NMET AND IMETR
C           - ADDED DETECTION OF SPECIAL PARENT
C             METASTABLE LINKS
C
C
C VERSION : 1.3
C DATE    : 10-02-97
C MODIFIED: HUGH SUMMERS
C     - INCLUDE UNSPECIFIED LOW N-SHELLS OF OUTER N-SHELL
C             IN PHFR ON ASSUMPTION THAT THEY ARE FILLED.
C
C VERSION : 1.4
C DATE    : 23-05-2003
C MODIFIED: Martin O'Mullane
C               - Pass in adf04 data rather than rewinding and
C                 reading it again.
C               - Remove all unused variables and reduced length of
```

```
C                 parameter list.
C               - Remove redundant code and format statements.
C
C VERSION : 1.5
C DATE    : 06-01-2004
C MODIFIED: Martin O'Mullane
C               - Pre-process configuration string with a new
C                 routine (ceprep) to account for leading d10 and
C                 f10-f14 terms.
C               - Add error trapping code to check for overruns
C                 and index=0 errors.
C
C VERSION : 1.6
C DATE    : 15-11-2004
C MODIFIED: Martin O'Mullane
C               - Increase to 3500 levels.
C
C VERSION : 1.7
C DATE    : 17-05-2007
C MODIFIED: Martin O'Mullane
C               - Updated comments as part of subroutine documentation
C                 procedure.
C-----------------------------------------------------------------------
      CHARACTER*18        CSTRGA(NDLEV)
      CHARACTER           EICHR(IODIMD)
      INTEGER             IA(NDLEV),   IGRPA(IODIMD),           IL
      INTEGER             ILA(NDLEV),  ILINK(NDMET,NDMET,2)
      INTEGER             ILPAR(NDMET),             IMETR(NDMET)
      INTEGER             INPAR(NDMET),             IODIMD
      INTEGER             IONLEV(NDMET,2),          IORB
      INTEGER             ISA(NDLEV),  ISPSYS(NDMET,2),         IZ
      INTEGER             IZ0,         IZ1
      INTEGER             IZETA4(NDMET,2,NDORB),    KGRPA(IODIMD)
      INTEGER             N0A(NDMET,2),             NDLEV,       NDMET
      INTEGER             NDORB,       NDQDN,       NDTHET,      NMET
      INTEGER             NORB,        NPMET,       NSYS(NDMET), NTHETA
      INTEGER             NZETA(NDMET,2)
      LOGICAL             LEICHR(IODIMD),           LEISS
      LOGICAL             LLINK(NDMET,NDMET,2)
      REAL*8              BWNO,        EIONA(NDMET,2,NDORB)
      REAL*8              ENPAR(NDMET),             PARMR(NDMET,2,4)
      REAL*8              QDORB((NDQDN*(NDQDN+1))/2)
      REAL*8              SBCHA(NDMET,2,NDTHET),    THETA(NDTHET)
      REAL*8              VORB(IODIMD),             WA(NDLEV)
      REAL*8              XITRUE(NDMET,2),          XJA(NDLEV)
```

5.38 d7bndl: Subroutine d7bndl from library adas4xx

```
      SUBROUTINE D7BNDL(NDLEV , NDTRN , NDMET , NDOSC ,
     +                  IL    , WA    , XJA   , NV    , TSCEF ,
     +                  ICNTE , IETRN , IE1A  , IE2A  , AA    , SCOM  ,
     +                  NMET  , IMETR , PTOT  , FMIN  , Z1    ,
     +                  ICTM  , IUMA  , CSTGMA,
     +                  FFMA  , WVMA  , GBMA  , PYMA
     +                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: D7BNDL *********************
C
C
C  PURPOSE:  SUBROUTINE TO DISCARD TRANSITIONS WITH AN OSCILLATOR
C            STRENGTH BELOW A CERTAIN INPUT VALUE. GETS SOME VALUES
C      VIA A PIPE FROM IDL AND THEN RETURNS SOME VALUES TO IDL
C      FOR DISPLAY USE IN CW_ADAS407_PROC
C
C
C  CALLING PROGRAM: D7ISPF
C
C
C  SUBROUTINE:
C
C
C
C  INPUT : (I*4)  NDLEV     = MAXIMUM NUMBER OF LEVELS THAT CAN BE READ
C  INPUT : (I*4)  NDTRN     = MAXIMUM NUMBER OF TRANSITIONS
C  INPUT : (I*4)  NDLEV     = MAXIMUM NUMBER OF LEVELS THAT CAN BE READ
C  INPUT : (I*4)  NDTRN     = MAXIMUM NUMBER OF TRANSITIONS
C  INPUT : (I*4)  NDMET     = MAXIMUM NUMBER OF METASTABLES
C  INPUT : (I*4)  NDOSC     = MAXIMUM NUMBER OF RADIATIVE TRANSITIONS
C                             ALLOWED FOR ASSEMBLING POWER FOR EACH
C                             METASTABLE
C
C  INPUT : (I*4)  IL        = NUMBER OF ENERGY LEVELS
C  INPUT : (R*8)  WA()      = ENERGY RELATIVE TO LEVEL 1 (CM-1)
C                             1ST DIM: LEVEL INDEX
C  INPUT : (R*8)  XJA()     = QUANTUM NUMBER (J-VALUE) FOR LEVEL 'IA()'
C                             (NOTE: (2*XJA)+1 = STATISTICAL WEIGHT)
C                             1ST DIM: LEVEL INDEX
C  INPUT : (I*4)  NV        = INPUT DATA FILE: NUMBER OF TEMPERATURES
C  INPUT : (R*8)  TSCEF(,)  = INPUT DATA FILE: ELECTRON TEMPERATURES
C                             2ND DIMENSION: 1 => KELVIN  (IFOUT=1)
C                                            2 => EV      (IFOUT=2)
C                                            3 => REDUCED (IFOUT=3)
C  INPUT : (I*4)  ICNTE     = NUMBER OF ELECTRON COLL. TRANSITIONS
C  INPUT : (I*4)  IETRN()   = ELECTRON IMPACT TRANSITION:
C                             INDEX VALUES IN MAIN TRANS. ARRAYS WHICH
C                             REPRESENT ELECTRON IMPACT TRANSITIONS.
C  INPUT : (I*4)  IE1A()    = TRANSITION: LOWER ENERGY LEVEL INDEX
C                             1ST DIM: ELECTRON COLL. TRANSITION INDEX
C  INPUT : (I*4)  IE2A()    = TRANSITION: UPPER ENERGY LEVEL INDEX
C                             1ST DIM: ELECTRON COLL. TRANSITION INDEX
C  INPUT : (R*8)  AA()      = TRANSITION: A-VALUE (SEC-1)
C                             1ST DIM: ELECTRON COLL. TRANSITION INDEX
C  INPUT : (R*8)  SCOM(,)   = TRANSITION:
C                             GAMMA VALUES             (CASE ' ' & 'P')
C                             RATE COEFFT. (CM3 SEC-1) (CASE 'H' & 'R')
```

```
C                               1ST DIMENSION - TEMPERATURE 'SCEF()'
C                               2ND DIMENSION - TRANSITION NUMBER
C
C  INPUT : (I*4)  NMET       = NUMBER OF METASTABLES SELECTED
C  INPUT : (I*4)  IMETR()     = INDEX OF METASTABLES IN LEVELE LIST
C                               1ST DIM: METASTABLE INDEX
C  OUTPUT: (R*8)  PTOT(,)     = TOTAL ZERO-DENS. RAD. POWER FOR EACH META.
C                               1ST DIM: SPECIFIC ION FILE TEMP. INDEX
C                               2ND DIM: METASTABLE INDEX
C  INPUT : (R*8)  FMIN        = MINIMIUM PERMITTED OSCILLATOR STRENGTH
C  INPUT : (R*8)  Z1          = RECOMBINING ION CHARGE
C
C
C  OUTPUT: (I*4)  ICTM()      = NUMBER OF INCLUDED TRANSITIONS FOR EACH
C                               METASTABLE
C                               1ST DIM: METASTABLE INDEX
C  OUTPUT: (I*4)  IUMA(,)     = INDEX OF METASTABLE ASSIGNED TRANSITION
C                               IN FULL ELECTRON COLL. TRANSITION LIST.
C                               1ST DIM: METASTABLE TRANSITION SET INDEX
C                               2ND DIM: METASTABLE INDEX
C  OUTPUT: (C*22) CSTGMA(,) = STRING IDENTIFIER FOR INCLUDED TRANSITION
C                               CONTAINING J,I,FIJ,WVLN
C                               1ST DIM: METASTABLE TRANSITION SET INDEX
C                               2ND DIM: METASTABLE INDEX
C  OUTPUT: (I*4)  FFMA(,)     = OSCILLATOR STRENGTH OF INCLUDED TRANSITION
C                               1ST DIM: METASTABLE TRANSITION SET INDEX
C                               2ND DIM: METASTABLE INDEX
C  OUTPUT: (I*4)  WVMA(,)     = WAVELENGTH (A)  OF INCLUDED TRANSITION
C                               1ST DIM: METASTABLE TRANSITION SET INDEX
C                               2ND DIM: METASTABLE INDEX
C  OUTPUT: (R*8)  GBMA(,,)    = GBAR FOR ALLLOWED TRANSITIONS
C                               1ST DIM: SPECIFIC ION FILE TEMP. INDEX
C                               2ND DIM: INCLUDED ALLOWED TRANS. INDEX
C                               3RD DIM: METASTABLE INDEX
C  OUTPUT: (R*8)  PYMA(,,)    = VAN REGEMORTER P FOR ALLOWED TRANSITIONS
C                               1ST DIM: SPECIFIC ION FILE TEMP. INDEX
C                               2ND DIM: INCLUDED ALLOWED TRANS. INDEX
C                               3RD DIM: METASTABLE INDEX
C
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          D7PYVR     ADAS      EVALUATES VAN REGEMORTER P FACTOR
C
C
C  AUTHOR: M O'MULLANE, UCC
C
C  DATE:    18/05/94
C
C  UPDATE:
C
C UNIX-IDL PORT:
C    WILLIAM OSBORN, TESSELLA SUPPORT SERVICES PLC.
C
C DATE:    22ND APRIL 1996
C
C VERSION: 1.1 DATE: 22-04-96
C MODIFIED: WILLIAM OSBORN
C     - FIRST VERSION.
C
```

```
C VERSION: 1.2 DATE: 22-04-96
C MODIFIED: WILLIAM OSBORN
C     REPLACED NDMET BY NMET IN LOOP TO READ IMETR

C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      CHARACTER*22        CSTGMA(NDOSC,NDMET)
      INTEGER             ICNTE,       ICTM(NDMET), IE1A(NDTRN)
      INTEGER             IE2A(NDTRN), IETRN(NDTRN),            IL
      INTEGER             IMETR(NDMET),             IUMA(NDOSC,NDMET)
      INTEGER             NDLEV,       NDMET,       NDOSC,       NDTRN
      INTEGER             NMET,        NV
      REAL*8              AA(NDTRN),   FFMA(NDOSC,NDMET),        FMIN
      REAL*8              GBMA(14,NDOSC,NDMET),     PTOT(14,NDMET)
      REAL*8              PYMA(14,NDOSC,NDMET),     SCOM(14,NDTRN)
      REAL*8              TSCEF(14,3), WA(NDLEV),   WVMA(NDOSC,NDMET)
      REAL*8              XJA(NDLEV),  Z1
```

5.39 d7clos: Subroutine d7clos from library adas4xx

```
      SUBROUTINE D7CLOS( IUNT11 , IUNT12 ,
     &                   OPEN11 , OPEN12 ,
     &                   IFCNT  , NDFILE , FILESL , FILESU ,
     &     DATE
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: D7CLOS *********************
C
C  PURPOSE: TO WRITE TERMINATOR SEQUENCES AND CLOSE FILES ON
C           UNIT11 (MAINCL FILE) AND UNIT12 (ATOMPARS FILE)
C
C  CALLING PROGRAM: ADAS407
C
C  SUBROUTINE:
C
C  INPUT : (I*4)   IUNT11   = UNIT NUMBER FOR MAINCL OUTPUT FILE
C  INPUT : (I*4)   IUNT12   = UNIT NUMBER FOR ATOMPARS OUTPUT FILE
C  INPUT : (L*4)   OPEN11   = .TRUE.  => MAINCL PASSING FILE OPENED
C                             .FALSE. => MAINCL PASSING FILE NOT OPENED
C  INPUT : (L*4)   OPEN12   = .TRUE.  => ATMPRS  PASSING FILE OPENED
C                             .FALSE. => ATMPRS  PASSING FILE NOT OPENED
C  INPUT : (I*4)   IFCNT    = NUMBER OF RECOMBINED/RECOMBINING FILE
C                             PAIRS USED
C  INPUT : (I*4)   NDFILE   = MAX. NUMBER OF FILES FOR DIMENSIONING FILESL
C       AND FILESU. THIS IS NECESSARY ON A DEC, BUT
C       NOT ON THE IBM AT JET (???)
C  INPUT : (C*80)  FILESL() = RECOMBINED FILE
C  INPUT : (C*80)  FILESU() = RECOMBINING FILE
C
C           (I*4)   I        = GENERAL INTEGER
C           (C*6)   USERID   = USER IDENTIFIER
C           (C*8)   DATE     = CURRENT DATE
C
C
C ROUTINES:
C           ROUTINE    SOURCE    BRIEF DESCRIPTION
C           ------------------------------------------------------------
C
C
C AUTHOR:  H. P. SUMMERS, JET
C          K1/1/57
C          JET EXT. 4941
C
C DATE:    12/07/94
C
C UNIX-IDL PORT:
C    WILLIAM OSBORN, TESSELLA SUPPORT SERVICES PLC.
C
C DATE:    22ND APRIL 1996
C
C VERSION: 1.1 DATE: 22-04-96
C MODIFIED: WILLIAM OSBORN
C     - FIRST VERSION.
C
C VERSION: 1.2 DATE: 23-04-96
C MODIFIED: WILLIAM OSBORN
C     INCREASED FILE NAME LENGTHS TO 80
```

```
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      CHARACTER*8         DATE
      CHARACTER*80        FILESL(NDFILE),           FILESU(NDFILE)
      INTEGER             IFCNT,       IUNT11,      IUNT12,      NDFILE
      LOGICAL             OPEN11,      OPEN12
```

5.40 d7cors: Subroutine d7cors from library adas4xx

```
       SUBROUTINE D7CORS(INZO,INZ,CORRS,CORRAD,CORRAR,CORRGA,LABEL)
       IMPLICIT REAL*8(A-H,O-Z)
C-----------------------------------------------------------------------
C      PURPOSE: CORRECT EX90 OUTPUT TO BETTER DATA. FOR A GIVEN
C      ISO-ELECTRONIC SEQUENCE, USES INTERPOLATION BETWEEN VALUES OF
C      Z0.
C
C      INPUT PARAMETERS:
C         INZO    -  ATOMIC MASS OF ION
C         INZ     -  CHARGE OF RECOMBINED ION
C      OUTPUT PARAMETERS:
C         CORRS   -  MULTIPLICITIVE CORRECTION FACTOR FOR IONISATION
C         CORRAD  -  MULTIPLICITIVE CORRECTION FACTOR FOR DI-EL. RECOM.
C         CORRAR  -  MULTIPLICITIVE CORRECTION FACTOR FOR RAD. RECOM.
C         CORRGA  -  MULTIPLICITIVE CORRECTION FACTOR FOR GA0 OF RAD.
C                    RECOM. APPROXIMATE FORM
C         LABEL   -  DETAILS OF CORRECTION DATA SOURCE
C
C      NOTES:
C        (1) SEQREP(ISEQ) RETURNS NO. OF ELECTRONS IN SEQUENCE ISEQ
C        (2) NZO IS MAXIMUM NO. OF ZO'S REPRESENTING A SEQUENCE
C        (3) NZOREP(ISEQ) RETURNS NO. OF ZO'S REPRESENTING A PARTICULAR
C            SEQUENCE
C        (4) ZOREP(ISEQ,IZO) CONTAINS SET OF REPRESENTATIVE ZO'S
C        (5) DATA FOR ZOREP = 1,100 IS DUMMY. SET EQUAL TO VALUES AT
C            THE PROPER ENDS OF THE DATA.  (E.G. AT ZO = 4,27)
C
C*****************  W.J.DICKSON JET 9/1/90 ****************************
C** PE BRIDEN 19AUG94 - MADE ALL REAL NUMBERS 8 BYTE INSTEAD OF 4 BYTE
C UNIX-IDL PORT:
C    WILLIAM OSBORN, TESSELLA SUPPORT SERVICES PLC.
C
C DATE:    22ND APRIL 1996
C
C VERSION: 1.1 DATE: 22-04-96
C MODIFIED: WILLIAM OSBORN
C     - FIRST VERSION. NO CHANGES TO IBM CODE.
C
C VERSION: 1.2 DATE: 16-05-07
C MODIFIED: ALLAN WHITEFORD
C     - UPDATED COMMENTS AS PART OF SUBROUTINE DOCUMENTATION
C       PROCEDURE
C-----------------------------------------------------------------------
      CHARACTER*40        LABEL
      INTEGER             INZ,         INZO
      REAL*8              CORRAD,      CORRAR,      CORRGA,      CORRS
```

5.41 d7data: Subroutine d7data from library adas4xx

```
       SUBROUTINE D7DATA( IUNIT  , NDLEV  , NDTRN ,
     &                    TITLED , IZ     , IZ0   , IZ1   , BWNO  ,
     &                    IL     ,
     &                    IA     , CSTRGA , ISA   , ILA   , XJA   , WA
     &                  )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: D7DATA *********************
C
C  PURPOSE:  TO FETCH LEVEL DATA FROM INPUT COPASE DATA SET.
C
C  CALLING PROGRAM: ADAS407
C
C  DATA:
C           THE UNITS USED IN THE DATA FILE ARE TAKEN AS FOLLOWS:
C
C           IONISATION POTENTIAL: WAVE NUMBER (CM-1)
C           INDEX LEVEL ENERGIES: WAVE NUMBER (CM-1)
C
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  IUNIT   = UNIT TO WHICH INPUT FILE IS ALLOCATED
C  INPUT : (I*4)  NDLEV   = MAXIMUM NUMBER OF LEVELS THAT CAN BE READ
C  INPUT : (I*4)  NDTRN   = MAX. NUMBER OF TRANSITIONS THAT CAN BE READ
C
C  OUTPUT: (C*3)  TITLED  = ELEMENT SYMBOL.
C  OUTPUT: (I*4)  IZ      =  RECOMBINED ION CHARGE READ
C  OUTPUT: (I*4)  IZ0     =         NUCLEAR CHARGE READ
C  OUTPUT: (I*4)  IZ1     = RECOMBINING ION CHARGE READ
C                           (NOTE: IZ1 SHOULD EQUAL IZ+1)
C  OUTPUT: (R*8)  BWNO    = IONISATION POTENTIAL (CM-1)
C
C  OUTPUT: (I*4)  IL      = INPUT DATA FILE: NUMBER OF ENERGY LEVELS
C
C  OUTPUT: (I*4)  IA()    = ENERGY LEVEL INDEX NUMBER
C  OUTPUT: (C*18) CSTRGA()= NOMENCLATURE/CONFIGURATION FOR LEVEL 'IA()'
C  OUTPUT: (I*4)  ISA()   = MULTIPLICITY FOR LEVEL 'IA()'
C                           NOTE: (ISA-1)/2 = QUANTUM NUMBER (S)
C  OUTPUT: (I*4)  ILA()   = QUANTUM NUMBER (L) FOR LEVEL 'IA()'
C  OUTPUT: (R*8)  XJA()   = QUANTUM NUMBER (J-VALUE) FOR LEVEL 'IA()'
C                           NOTE: (2*XJA)+1 = STATISTICAL WEIGHT
C  OUTPUT: (R*8)  WA()    = ENERGY RELATIVE TO LEVEL 1 (CM-1) FOR LEVEL
C                           'IA()'
C
C
C          (I*4)  I4UNIT  = FUNCTION (SEE ROUTINE SECTION BELOW)
C          (I*4)  I       = GENERAL USE.
C          (I*4)  IABT    = RETURN CODE FROM 'R8FCTN' (0 => NO ERROR)
C                           OR FROM INTERROGATION OF 'C7'
C          (I*4)  IFIRST  = BYTE POSITION OF START OF NUMBER IN BUFFER
C          (I*4)  ILAST   = BYTE POSITION OF END   OF NUMBER IN BUFFER
C          (I*4)  IWORD   = THE WORD POSITION OF THE REQUIRED DATA IN
C                           A STRING TO BE INTERROGATED BY XXWORD.
C          (I*4)  J       = GENERAL USE.
C          (I*4)  LENCST  = BYTE LENGTH OF STRING CSTRGA()
C          (I*4)  NWORDS  = NUMBER OF NUMBERS STORED IN BUFFER
C          (I*4)  ILINE   = ENERGY LEVEL INDEX FOR CURRENT LINE
```

```
C          (I*4)  IRECL   = RECORD LENGTH OF INPUT DATASET (<=128)
C
C          (C*7)  C7      = USED TO PARSE VALUE FOR XJA()
C          (C*7)  CDELIM  = DELIMITERS FOR INPUT OF DATA FROM HEADERS
C          (C*18) C18     = USED TO PARSE VALUE TO CSTRGA()
C          (C*80) CLINE   = CURRENT ENERGY LEVEL INDEX PARAMETER LINE
C          (C*128)BUFFER  = GENERAL STRING BUFFER STORAGE
C
C          (L*4)  LDATA   = IDENTIFIES  WHETHER  THE END OF AN  INPUT
C                           SECTION IN THE DATA SET HAS BEEN LOCATED.
C                           (.TRUE. => END OF SECTION REACHED)
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          XXWORD     ADAS      EXTRACT POSITION OF NUMBER IN BUFFER
C          I4UNIT     ADAS      FETCH UNIT NUMBER FOR OUTPUT OF MESSAGES
C          R8FCTN     ADAS      CONVERTS FROM CHARACTER TO REAL VARIABLE
C
C
C ROUTINES: NONE
C
C AUTHOR:  H. P. SUMMERS, JET
C          K1/1/47
C          JET EXT. 4941
C
C DATE:    19/05/94
C
C UPDATE:  12/07/94 - H. P. SUMMERS - ALLOWED DUMMY BARE NUCLEUS FILE
C                                     TO BE READ BY DETECTING IZ0=IZ
C
C UNIX-IDL PORT:
C    WILLIAM OSBORN, TESSELLA SUPPORT SERVICES PLC.
C
C DATE:    25TH MARCH 1996
C
C VERSION: 1.1 DATE: 25-03-96
C MODIFIED: WILLIAM OSBORN
C     - FIRST VERSION.
C
C VERSION: 1.2 DATE: 09-09-96
C MODIFIED: WILLIAM OSBORN / PAUL BRIDEN
C           - INSTEAD OF USING FORMAT SPECIFIER F15.0 WHEN
C             INTERNALLY READING A FLOATING POINT NUMBER,
C             CREATE THE APPROPRIATE SPECIFIER WITHIN CFORM7
C             AND USE THIS.
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      CHARACTER*(*)       CSTRGA(NDLEV)
      CHARACTER*3         TITLED
      INTEGER             IA(NDLEV),   IL,          ILA(NDLEV)
      INTEGER             ISA(NDLEV),  IUNIT,       IZ,          IZ0
      INTEGER             IZ1,         NDLEV,       NDTRN
      REAL*8              BWNO,        WA(NDLEV),   XJA(NDLEV)
```

5.42 d7datr: Subroutine d7datr from library adas4xx

```
C
       SUBROUTINE D7DATR( IUNIT  , NDLEV  , NDTRN , NDQDN ,
     &                    TITLED , IZ     , IZ0   , IZ1   , BWNO  ,
     &                    IL     , QDORB  , LQDORB, QDN   ,
     &                    IA     , CSTRGA , ISA   , ILA   , XJA   , WA ,
     &                    NV     , SCEF   ,
     &                    ITRAN  , MAXLEV ,
     &                    TCODE  , I1A    , I2A   , AVAL  , SCOM
     &                  )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: D7DATR *********************
C
C  PURPOSE:  TO FETCH DATA FROM INPUT COPASE DATA SET.  THE SUBROUTINE
C            IS AN EXTENSION TO BXDATA TO OBTAIN  ORBITAL QUANTUM
C            DEFECTS.  IN ALL OTHER RESPECTS IT IS IDENTICAL TO BXDATA.
C
C  CALLING PROGRAM: ADAS407
C
C  DATA:
C           THE 'REAL' DATA IN THE FILE IS REPRESENTED IN AN ABBREVIATED
C           FORM WHICH OMITS THE "D" OR "E" EXPONENT SPECIFIER.
C           e.g. 1.23D-06 or 1.23E-06 IS REPRESENTED AS 1.23-06
C                6.75D+07 or 6.75E+07 IS REPRESENTED AS 6.75+07
C
C           THEREFORE THE FORM OF EACH 'REAL' NUMBER IN THE DATA SET IS:
C                          N.NN+NN or N.NN-NN
C
C           THE UNITS USED IN THE DATA FILE ARE TAKEN AS FOLLOWS:
C
C           IONISATION POTENTIAL: WAVE NUMBER (CM-1)
C           INDEX LEVEL ENERGIES: WAVE NUMBER (CM-1)
C           TEMPERATURES        : KELVIN
C           A-VALUES            : SEC-1
C           GAMMA-VALUES        :
C           RATE COEFFT.        : CM3 SEC-1
C
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  IUNIT   = UNIT TO WHICH INPUT FILE IS ALLOCATED
C  INPUT : (I*4)  NDLEV   = MAXIMUM NUMBER OF LEVELS THAT CAN BE READ
C  INPUT : (I*4)  NDTRN   = MAX. NUMBER OF TRANSITIONS THAT CAN BE READ
C  INPUT : (I*4)  NDQDN   = MAX. NUMBER OF N-SHELLS FOR QUANTUM DEFECTS
C
C  OUTPUT: (C*3)  TITLED  = ELEMENT SYMBOL.
C  OUTPUT: (I*4)  IZ      =  RECOMBINED ION CHARGE READ
C  OUTPUT: (I*4)  IZ0     =         NUCLEAR CHARGE READ
C  OUTPUT: (I*4)  IZ1     = RECOMBINING ION CHARGE READ
C                           (NOTE: IZ1 SHOULD EQUAL IZ+1)
C  OUTPUT: (R*8)  BWNO    = IONISATION POTENTIAL (CM-1)
C
C  OUTPUT: (I*4)  IL      = INPUT DATA FILE: NUMBER OF ENERGY LEVELS
C  OUTPUT: (R*8)  QDORB() = QUANTUM DEFECTS FOR ORBITALS
C                           1ST DIM: INDEX FOR NL ORBITAL (CF INDX)
C  OUTPUT: (L*4)  LQDORB()= .TRUE.  => SOURCE DATA AVAILABLE FOR QD.
C                         = .FALSE. => SOURCE DATA NOT AVAILABE QD.=0.0
C  OUTPUT: (R*8)  QDN()   = QUANTUM DEFECT FOR N-SHELLS.  NON-ZERO ONLY
C                           FOR ADF04 FILES WITH ORBITAL ENERGY DATA
```

```
C                            1ST. DIM: N-SHELL (1<=N<=NDQDN)
C
C  OUTPUT: (I*4)  IA()    = ENERGY LEVEL INDEX NUMBER
C  OUTPUT: (C*18) CSTRGA()= NOMENCLATURE/CONFIGURATION FOR LEVEL 'IA()'
C  OUTPUT: (I*4)  ISA()   = MULTIPLICITY FOR LEVEL 'IA()'
C                           NOTE: (ISA-1)/2 = QUANTUM NUMBER (S)
C  OUTPUT: (I*4)  ILA()   = QUANTUM NUMBER (L) FOR LEVEL 'IA()'
C  OUTPUT: (R*8)  XJA()   = QUANTUM NUMBER (J-VALUE) FOR LEVEL 'IA()'
C                           NOTE: (2*XJA)+1 = STATISTICAL WEIGHT
C  OUTPUT: (R*8)  WA()    = ENERGY RELATIVE TO LEVEL 1 (CM-1) FOR LEVEL
C                           'IA()'
C
C  OUTPUT: (I*4)  NV      = INPUT DATA FILE: NUMBER OF GAMMA/TEMPERATURE
C                           PAIRS FOR A GIVEN TRANSITION.
C  OUTPUT: (R*8)  SCEF()  = INPUT DATA FILE: ELECTRON TEMPERATURES (K)
C                           (INITIALLY JUST THE MANTISSA. SEE 'ITPOW()')
C                           (NOTE: TE=TP=TH IS ASSUMED)
C
C  OUTPUT: (I*4)  ITRAN   = INPUT DATA FILE: NUMBER OF TRANSITIONS
C  OUTPUT: (I*4)  MAXLEV  = HIGHEST INDEX LEVEL IN READ TRANSITIONS
C
C  OUTPUT: (C*1)  TCODE() = TRANSITION: DATA TYPE POINTER:
C                           ' ' => Electron Impact   Transition
C                           'P' => Proton   Impact   Transition
C                           'H' => Charge   Exchange Recombination
C                           'R' => Free     Electron Recombination
C  OUTPUT: (I*4)  I1A()   = TRANSITION:
C                            LOWER ENERGY LEVEL INDEX (CASE ' ' & 'P')
C                            SIGNED PARENT      INDEX (CASE 'H' & 'R')
C  OUTPUT: (I*4)  I2A()   = TRANSITION:
C                            UPPER ENERGY LEVEL INDEX (CASE ' ' & 'P')
C                            CAPTURING    LEVEL INDEX (CASE 'H' & 'R')
C  OUTPUT: (R*8)  AVAL()  = TRANSITION:
C                            A-VALUE (SEC-1)          (CASE ' ')
C                            NEUTRAL BEAM ENERGY      (CASE 'H')
C                            NOT USED                 (CASE 'P' & 'R')
C  OUTPUT: (R*8)  SCOM(,) = TRANSITION:
C                            GAMMA VALUES             (CASE ' ' & 'P')
C                            RATE COEFFT. (CM3 SEC-1) (CASE 'H' & 'R')
C                           1ST DIMENSION - TEMPERATURE 'SCEF()'
C                           2ND DIMENSION - TRANSITION NUMBER
C
C          (I*4)  NVMAX   = PARAMETER = MAX. NUMBER OF TEMPERATURES
C                                       THAT CAN BE READ IN.
C          (I*4)  MTIED   = PARAMETER = MUST BE GREATER THAN OR EQUAL TO
C                                       THE MAX. NO. OF LEVELS.
C          (R*8)  DZERO   = PARAMETER = MINIMUM VALUE FOR 'AVAL()' AND
C                                       'SCOM()' ARRAYS = 1.0D-30
C
C          (I*4)  I4UNIT  = FUNCTION (SEE ROUTINE SECTION BELOW)
C          (I*4)  IQS     = X-SECT DATA FORMAT SELECTOR
C                           NOTE: IQS=3 ONLY ALLOWED IN THIS PROGRAM
C          (I*4)  I       = GENERAL USE.
C          (I*4)  IABT    = RETURN CODE FROM 'R8FCTN' (0 => NO ERROR)
C                           OR FROM INTERROGATION OF 'C7'
C          (I*4)  IFIRST  = BYTE POSITION OF START OF NUMBER IN BUFFER
C          (I*4)  ILAST   = BYTE POSITION OF END   OF NUMBER IN BUFFER
C          (I*4)  INDX    = INDEXING FUNCTION FOR NL ORBITALS IN
C                           QDORB() AS INDX(N,L)
C          (I*4)  IORB    = ORBITAL INDEX
C          (I*4)  IWORD   = THE WORD POSITION OF THE REQUIRED DATA IN
```

```
C                           A STRING TO BE INTERROGATED BY XXWORD.
C          (I*4)  J       = GENERAL USE.
C          (I*4)  J1      = INPUT DATA FILE - SELECTED TRANSITION:
C                            LOWER ENERGY LEVEL INDEX (CASE ' ' & 'P')
C          (I*4)  J2      = INPUT DATA FILE - SELECTED TRANSITION:
C                            UPPER ENERGY LEVEL INDEX (CASE ' ' & 'P')
C                            CAPTURING    LEVEL INDEX (CASE 'H' & 'R')
C          (I*4)  K       = GENERAL USE
C          (I*4)  L       = GENERAL USE FOR ORBITAL L
C          (I*4)  L1      = GENERAL USE
C          (I*4)  LENCST  = BYTE LENGTH OF STRING CSTRGA()
C          (R*8)  NLAST   = N-SHELL MARKER FOR ORBITALS
C          (I*4)  NWORDS  = NUMBER OF NUMBERS STORED IN BUFFER
C          (I*4)  ILINE   = ENERGY LEVEL INDEX FOR CURRENT LINE
C          (I*4)  IAPOW   = EXPONENT OF 'AVALM'
C          (I*4)  IGPOW() = EXPONENT OF 'GAMMA()'
C          (I*4)  ITPOW() = TEMPERATURES - EXPONENT
C                           NOTE: MANTISSA INITIALLY KEPT IN 'SCEF()'
C
C          (R*4)  ZF      = SHOULD BE EQUIVALENT TO 'IZ1'
C
C          (R*8)  AVALM   = INPUT DATA FILE - SELECTED TRANSITION:
C                           MANTISSA OF:   ('IAPOW' => EXPONENT)
C                            A-VALUE (SEC-1)          (CASE ' ')
C                            NEUTRAL BEAM ENERGY      (CASE 'H')
C                            NOT USED                 (CASE 'P' & 'R')
C          (R*8)  GAMMA() = INPUT DATA FILE - SELECTED TRANSITION:
C                           MANTISSA OF: ('IGPOW()' => EXPONENT)
C                            GAMMA VALUES             (CASE ' ' & 'P')
C                            RATE COEFFT. (CM3 SEC-1) (CASE 'H' & 'R')
C                           DIMENSION => TEMPERATURE 'SCEF()'
C          (R*8)  QDORB() = QUANTUM DEFECTS FOR ORBITALS
C                           1ST DIM: INDEX FOR NL ORBITAL (CF INDX)
C
C          (C*7)  C7      = USED TO PARSE VALUE FOR XJA()
C          (C*7)  CDELIM  = DELIMITERS FOR INPUT OF DATA FROM HEADERS
C          (C*18) C18     = USED TO PARSE VALUE TO CSTRGA()
C          (C*80) CLINE   = CURRENT ENERGY LEVEL INDEX PARAMETER LINE
C          (C*128)BUFFER  = GENERAL STRING BUFFER STORAGE
C          (C*3)  CITPOW()= USED TO PARSE VALUES TO ITPOW()
C          (C*5)  CSCEF() = USED TO PARSE VALUES TO SCEF()
C          (C*7)  CFORM7  = FORMAT FOR INTERNAL READING OF REAL NUMBER
C
C          (L*4)  LDATA   = IDENTIFIES  WHETHER  THE END OF AN  INPUT
C                           SECTION IN THE DATA SET HAS BEEN LOCATED.
C                           (.TRUE. => END OF SECTION REACHED)
C          (L*4)  LTCHR   = .TRUE.  => CURRENT 'TCODE()' = 'H' OR 'R'.
C                         = .FALSE. => CURRENT 'TCODE()'.NE.'H' OR 'R'.
C          (L*4)  LTCPR   = .TRUE.  => CURRENT 'TCODE()' = 'P' OR 'R'.
C                         = .FALSE. => CURRENT 'TCODE()'.NE.'P' OR 'R'.
C          (L*4)  LERROR  = .TRUE.  => UNTIED LEVEL FOUND
C                         = .FALSE. => ALL LEVELS TIED
C          (L*4)  LTIED() = .TRUE.  => SPECIFIED LEVEL TIED
C                         = .FALSE. => SPECIFIED LEVEL IS UNTIED
C                           DIMENSION => LEVEL INDEX
C          (L*4)  LQDORB()= .TRUE.  => SOURCE DATA AVAILABLE FOR QD.
C                         = .FALSE. => SOURCE DATA NOT AVAILABE QD.=0.0
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
```

```
C          XXWORD     ADAS      EXTRACT POSITION OF NUMBER IN BUFFER
C          I4UNIT     ADAS      FETCH UNIT NUMBER FOR OUTPUT OF MESSAGES
C          R8FCTN     ADAS      CONVERTS FROM CHARACTER TO REAL VARIABLE
C          INDX       INTRINSIC INDEXES NL ORBITAL IN QDORB() ARRAY
C
C NOTE:             LTCHR        LTCPR         TCODE()
C                  ----------------------------------
C                  .TRUE.       .TRUE.    =>     'R'
C                  .TRUE.       .FALSE.   =>     'H'
C                  .FALSE.      .TRUE.    =>     'P'
C                  .FALSE.      .FALSE.   =>     ' '
C
C        FOR A-VALUES & GAMMA-VALUES ENTRIES LESS THAN 'DZERO' ARE TAKEN
C        AS BEING EQUAL TO DZERO. THIS AFFECTS THE 'AVAL()' AND 'SCOM()'
C        ARRAYS.
C
C
C AUTHOR:  HUGH P. SUMMERS, UNIVERSITY OF STRATHCLYDE
C          JA8.08
C          TEL. 0141-553-4196
C
C DATE:    13/08/97
C
C UPDATE:
C
C VERSION: 1.1 DATE: 25-02-98
C MODIFIED: RICHARD MARTIN
C - PUT UNDER SCCS CONTROL.
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      CHARACTER*(*)       CSTRGA(NDLEV)
      CHARACTER           TCODE(NDTRN)
      CHARACTER*3         TITLED
      INTEGER             I1A(NDTRN),  I2A(NDTRN),  IA(NDLEV),   IL
      INTEGER             ILA(NDLEV),  ISA(NDLEV),  ITRAN,       IUNIT
      INTEGER             IZ,          IZ0,         IZ1,         MAXLEV
      INTEGER             NDLEV,       NDQDN,       NDTRN,       NV
      LOGICAL             LQDORB((NDQDN*(NDQDN+1))/2)
      REAL*8              AVAL(NDTRN), BWNO,        QDN(NDQDN)
      REAL*8              QDORB((NDQDN*(NDQDN+1))/2)
      REAL*8              SCEF(NVMAX), SCOM(NVMAX,NDTRN)
      REAL*8              WA(NDLEV),   XJA(NDLEV)
```

5.43 d7exps: Subroutine d7exps from library adas4xx

```
      SUBROUTINE D7EXPS( NDMET   , NDCONF , NDTHET , NDORB  , NDLEV   ,
     &                   ndtrn   , ndqdn  , LTADJ  ,
     &                   IZ      , IZ0    , IZ1    ,
     &                   il      , ia     , isa    , ila    , xja     ,
     &                   cstrga  , wa     , bwno   , iorb   , qdorb   ,
     &                   ipl     , ipa    , ipsa   , ipla   , xpja    ,
     &                   cstrgpa , wpa    , bwnop  ,
     &                   icnte2  , ie1a2  , ie2a2  , aval2  ,
     &                   NTHETA  , THETA  , ITSELA , NPMET  , IPMETR  ,
     &                   ISPRT   , ISPSYS , NSYS   , INPAR  , ILPAR   ,
     &                   ENPAR   , TRMPRT , SPNFAC ,
     &                   NORB    , VORB   , LEICHR ,
     &                   CI4     , EIONA  , IZETA4 , NZETA  , SAO     ,
     &                   IONLEV  , XITRUE , NCUT   , N0A    , PARMR   ,
     &                   ALFRA   , ALFRA0 , ALFRAR , NCONFG ,
     &                   WVMIN   , WVMAX  ,
     &                   ECF     , FCF    , PCF    , WCF    , W       ,
     &                   NCF     , LCF    , NDCF   , LDCF   , NDMIN   ,
     &                   E       , DE0    , DE     , FM0    , FM      ,
     &                   IINAA   , IIPNAA , NCTAA  , NCTAAC , ECTAA   ,
     &                   NTRANS  , ITYPEA , N1A    , NCUTT  , PARMD   ,
     &                   EDISGP  , SCALGP , ADIELO , ALFDA  , ALFPART ,
     &                   LLINK   , ILINK  , LEISS  , NMET   , IMETR
     &                 )

C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: D7EXPS *********************
C
C
C  PURPOSE:
C           (1) GENERATES APPROXIMATE FORM PARAMETERS AND NUMERICAL
C               VALUES FOR IONISATION AND RECOMBINATION RATES FROM
C               SPECIFIC ION FILES
C           (2) RETURNS DATA REQUIRED FOR A MAINCL INPUT FILE RESOLVED
C               INTO PARENT/SPIN SYSTEM COMPONENTS.
C           (3) RETURNS PARAMETERS REQUIRED FOR AN ATOMPARS FILE
C
C  CALLING PROGRAM: ADAS407
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  IUNIT    = UNIT NUMBER FOR SPECIFIC ION FILE FOR
C                            RECOMBINED ION
C  INPUT : (I*4)  IUNIT1   = UNIT NUMBER FOR SPECIFIC ION FILE FOR
C                            RECOMBINING ION
C  INPUT : (I*4)  NDMET    = MAXIMUM NUMBER OF METASTABLES ALLOWED
C  INPUT : (I*4)  NDTHET   = MAXIMUM NUMBER OF TEMPS. FOR MAINCL FILE
C  INPUT : (I*4)  NDCONF   = MAXIMUM NUMBER OF CONFIGURATIONS ALLOWED
C  INPUT : (I*4)  NDORB    = MAXIMUM NUMBER OF ELECTRON ORBITALS
C
C  INPUT : (L*4)   LTADJ   = .TRUE. => ADJUST PARMS FROM SPECIAL TABLES
C                            .FALSE.=> DO NOT ADJUST PARMS FROM TABLES
C
C  INPUT : (I*4)  IZ       = RECOMBINED ION CHARGE
C  INPUT : (I*4)  IZ1      = RECOMBINING ION CHARGE
C
C  INPUT : (I*4)  NTHETA   = NUMBER OF TEMPERATURES FOR MAINCL FILE
```

```
C  INPUT : (R*8)  THETA()  = Z-SCALED TEMPERATURES FOR MAINCL FILE
C  INPUT : (I*4)  ITSELA() = TEMPERATURE INDEX FOR POWER MATCHING
C                            1ST IND: RECOMBINED ION METASTABLE INDEX
C
C  INPUT : (I*4)  NPMET    = NO. OF RECOMBINING ION (PARENT) METASTABLES
C  INPUT : (I*4)  IPMETR() = INDICES OF RECOMBINING ION (PARENT)
C                            METASTABLES IN LEVEL LIST
C  INPUT : (L*4)  LLINK(,,) = .TRUE.  => LINK EXISTS
C                             .FALSE. => NO LINK EXISTS
C                              1ST DIM: METASTABLE INDEX
C                              2ND DIM: PARENT METASTABLE INDEX
C                              3RD DIM: SPEN SYSTEM INDEX
C  INPUT : (L*4)  ILINK(,,) = DECIMAL ORBITAL INDEX FOR RECOMBINED
C                             ION ORBITAL DIFFERENCE WITH PARENT
C                              1ST DIM: METASTABLE INDEX
C                              2ND DIM: PARENT METASTABLE INDEX
C                              3RD DIM: SPEN SYSTEM INDEX
C  INPUT : (L*4)  LEISS    = .TRUE. => PARENTS AND METASTABLES FOUND
C                                      TO HAVE EISSNER CONFIG. FORMS
C                            .FALSE => NOT EISSNER CONFIG. FORMS
C  INPUT : (I*4)  NMET     = NUMBER OF METASTABLES (1 <= NMET <= 'NDMET')
C  INPUT : (I*4)  IMETR()  = INDEX OF METASTABLE IN COMPLETE LEVEL LIST
C                           (ARRAY SIZE = 'NDMET' )
C
C  OUTPUT: (I*4)  ISPRT()  = RECOMBINING ION (PARENT) SPIN
C                            1ST DIM: PARENT INDEX
C  OUTPUT: (I*4)  ISPSYS(,)= RECOMBINED ION SPIN
C                            1ST DIM: PARENT INDEX
C                            2ND IND: SPIN SYSTEM INDEX
C  OUTPUT: (I*4)  NSYS()   = NUMBER OF SPIN SYSTEMS FOR RECOMBINED ION
C                            1ST DIM: PARENT INDEX
C  OUTPUT: (I*4)  INPAR()  = N QUANTUM NO. SUM FOR ELECTRONS OF PARENT
C                            1ST DIM: PARENT INDEX
C  OUTPUT: (I*4)  ILPAR()  = L QUANTUM NO. SUM FOR ELECTRONS OF PARENT
C                            1ST DIM: PARENT INDEX
C  OUTPUT: (R*8)  ENPAR()  = RECOMBINING ION (PARENT) ENERGY
C                            1ST DIM: PARENT INDEX
C  OUTPUT: (C*2)  TRMPRT() = RECOMBINING ION METASTABLE (PARENT) TERM
C                            1ST DIM: PARENT INDEX
C  OUTPUT: (I*4)  NCUT()   = N-SHELL AUTOIONISATION CUT-OFF FOR PARENT
C                            1ST DIM: PARENT INDEX
C  OUTPUT: (I*4)  N0A(,)   = LOWEST ALLOWED N-SHELL
C                            1ST DIM: PARENT INDEX
C                            2ND DIM: SPIN SYSTEM INDEX
C  OUTPUT: (R*4)  PARMR(,,)= PARAMETERS OF RADIATIVE RECOMBINATION
C                            APPROXIMATE FORMS
C                            1ST DIM: PARENT INDEX
C                            2ND DIM: SPIN SYSTEM INDEX
C                            3RD DIM: PARMS.  1: EFF. N FOR LOWEST LEVEL
C                                             2: PHASE SPACE FACTOR
C                                             3: ENERGY DISPLACEMENT
C                                             4: SCALING MULTIPLIER
C
C          (I*4)  ITDIMD   = PARAMETER = LIMIT NUMBER OF TEMPERATURES
C                                        INTRINSIC TO ROUTINE
C          (I*4)  IMDIMD   = PARAMETER = LIMIT NUMBER OF METASTABLES
C                                        INTRINSIC TO ROUTINE
C          (I*4)  IODIMD   = PARAMETER = LIMIT NUMBER OF ELEC. ORBITALS
C          (R*8)  SPNFAC(,)= SPIN WEIGHT FRACTION FOR PARENT/SPIN SYSTEM
C                            1ST DIM: PARENT INDEX
C                            2ND DIM: RECOMBINED ION SPIN SYSTEM INDEX
```

```
C          (R*8)  SPNSUM() = SPIN SYSTEM WEIGHT SUM BASED ON PARENT
C                            1ST DIM: PARENT INDEX
C          (I*4)  IGRPA()  = NUMBER OF ELECTRONS ALLOWED IN EACH SHELL
C                            1ST DIM: SHELL INDEX (1=1S, 2=2S ETC)
C          (C*1)  EICHR()  = EISSNER NOTATION CHARACTER FOR ORBITAL
C                            1ST DIM: ORBITAL INDEX
C          (L*4)  LEICHR() = .TRUE. => EISSNER ORBITAL USED
C                            .FALSE => EISSNER ORBITAL NOT USED
C          (I*4)  KGRPA()  = NUMBER OF ELECTRONS IN EACH SHELL
C                            1ST DIM: SHELL INDEX (1=1S, 2=2S ETC)
C          (R*8)  VORB()   = EFFECT. PRINC. QUANT. NO. FOR ORBITAL
C                            1ST DIM: SHELL INDEX (1=1S, 2=2S ETC)
C          (R*8)  EPSIL()  = ENERGY OF ORBITAL (RYDBERG)
C                            1ST DIM: SHELL INDEX (1=1S, 2=2S ETC)
C          (R*8)  SAO(,,)  = BEST EXTIMATE OF METASTABLE AVERAGED
C                            IONISATION RATE
C                            1ST DIM: PARENT INDEX
C                            2ND IND: SPIN SYSTEM INDEX
C                            3RD IND: TEMPERATURE INDEX
C          (C*1)  CHLA()   = CONVERTS NUMERICAL VALUE FOR L QUANTUM
C                            TO CHARACTER VALUE (CAPITAL).  NOTE
C                            THAT L+1 (<11) IS THE CALL PARAMETER.
C          (I*4)  ILPRT()  = RECOMBINING ION (PARENT) TOTAL ORBITAL
C                            ANGULAR MOMENTUM
C                            1ST DIM: PARENT INDEX
C
C NOTE:
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          D7AUTS     ADAS      ANALYSES FOR IONISATION RATE PARAMETERS
C          D7ALFS     ADAS      ANALYSES FOR RECOMBINATION PARAMETERS
C
C AUTHOR:  H. P. SUMMERS, JET
C          K1/1/57
C          JET EXT. 4941
C
C DATE:    27/06/94
C
C UPDATE:  04/07/95 - HPS  CORRECTED ERROR BARE NUCLEUS CASE TO ENSURE
C                          NSHEL=1, NEL=1 CHEISA(1)='1' SET.
C                          NOTE:  INSUM, ILSUM ARE NOT SATISFACTORY IN
C                          THE BARE NUCLEUS CASE.  STILL TO RECONSIDER
C                          THE ALGORITM FOR DECIDING INNER CLOSED SHELLS
C                          BARE NUCLEUS MAKES A FALSE ASSUMPTION BUT
C                          WITHOUT SERIOUS CONSEQUENCE.
C UPDATE:  07/03/96 - HPS  REMOVED VALUE ASSIGNMENT OF NTHETA
C                          REMOVED VALUE ASSIGNMENT OF MAXDTA
C
C UNIX-IDL PORT:
C    WILLIAM OSBORN, TESSELLA SUPPORT SERVICES PLC.
C
C DATE:    22ND APRIL 1996
C
C VERSION: 1.1 DATE: 22-04-96
C MODIFIED: WILLIAM OSBORN
C     - FIRST VERSION.
C
C VERSION: 1.2 DATE: 14-05-96
C MODIFIED: WILLIAM OSBORN
```

```
C     REARRANGED ARGUMENTS TO STAY UNDER
C     LIMIT OF 20 CONTINUATION CHARACTERS AT ARCETRI AND GARCHING
C VERSION: 1.3 DATE: 20-08-96
C MODIFIED: HUGH SUMMERS + WILLIAM OSBORN
C     - CORRECTED ASSIGNMENT OF 'TRMPRT'
C           - ADDED FOLLOWING TO CALL PARAMETERS
C             LLINK,ILINK,LEISS,NMET AND IMETR
C VERSION: 1.4 DATE: 24-09-96
C MODIFIED: HUGH SUMMERS + WILLIAM OSBORN
C     - INTRODUCED ILPRT AND CORRECT OUTPUT TRMPRT
C
C
C VERSION : 1.5
C DATE    : 23-05-2003
C MODIFIED: Martin O'Mullane
C               - Pass through adf04 data for d7alfs and d7auts.
C               - Do not rewind files to get parent data; use
C                 the new arguments.
C               - Make implicit none.
C               - Remove all unused variables and reduced length of
C                 parameter list.
C               - Remove redundant code and format statements.
C
C VERSION : 1.6
C DATE    : 04-11-2003
C MODIFIED: Hugh Summers
C               - checked iodimd consistency with passed ndorb.
C               - Extended igrpa eichr,chla
C               - corrected array indexing error in applying corrad
C                 to parmd(ipar,6,j)
C
C VERSION : 1.7
C DATE    : 06-01-2004
C MODIFIED: Martin O'Mullane
C               - Remove redundant nia, lia, wia, nja, lja and wja
C                 variables.
C               - Pre-process configuration string with a new
C                 routine (ceprep) to account for leading d10 and
C                 f10-f14 terms.
C               - Add error trapping code to check for overruns
C                 and index=0 errors.
C
C-----------------------------------------------------------------------
      CHARACTER*18        CSTRGA(NDLEV),            CSTRGPA(NDLEV)
      CHARACTER*2         TRMPRT(NDMET)
      INTEGER             IA(NDLEV),   ICNTE2,      IE1A2(NDTRN)
      INTEGER             IE2A2(NDTRN),             IINAA(NDMET,NDCONF)
      INTEGER             IIPNAA(NDMET,NDCONF),     IL
      INTEGER             ILA(NDLEV),  ILINK(NDMET,NDMET,2)
      INTEGER             ILPAR(NDMET),             IMETR(NDMET)
      INTEGER             INPAR(NDMET),             IONLEV(NDMET,2)
      INTEGER             IORB,        IPA(NDLEV),  IPL
      INTEGER             IPLA(NDLEV), IPMETR(NDMET)
      INTEGER             IPSA(NDLEV), ISA(NDLEV),  ISPRT(NDMET)
      INTEGER             ISPSYS(NDMET,2),          ITSELA(NDMET)
      INTEGER             ITYPEA(NDMET,NDCONF),     IZ,          IZ0
      INTEGER             IZ1,         IZETA4(NDMET,2,NDORB)
      INTEGER             LCF(NDMET,NDCONF),        LDCF(NDMET,NDCONF)
      INTEGER             N0A(NDMET,2),             N1A(NDMET,NDCONF)
      INTEGER             NCF(NDMET,NDCONF),        NCONFG
      INTEGER             NCTAA(NDMET,NDCONF),      NCTAAC(NDMET,NDCONF)
```

```
INTEGER            NCUT(NDMET), NCUTT(NDMET,NDCONF)
INTEGER            NDCF(NDMET,NDCONF),       NDCONF,       NDLEV
INTEGER            NDMET,        NDMIN(NDMET),             NDORB
INTEGER            NDQDN,        NDTHET,       NDTRN,        NMET
INTEGER            NORB,         NPMET,        NSYS(NDMET), NTHETA
INTEGER            NTRANS(NDMET),              NZETA(NDMET,2)
LOGICAL            LEICHR(NDORB),              LEISS
LOGICAL            LLINK(NDMET,NDMET,2),       LTADJ
REAL*8             ADIELO(NDMET,2,NDTHET),   ALFDA(NDMET,NDTHET)
REAL*8             ALFPART(NDMET,NDCONF,NDTHET)
REAL*8             ALFRA(NDMET,2,NDTHET)
REAL*8             ALFRA0(NDMET,2,NDTHET)
REAL*8             ALFRAR(NDMET,2,NDTHET),   AVAL2(NDTRN)
REAL*8             BWNO,         BWNOP,        CI4
REAL*8             DE(NDMET),    DE0(NDMET),   E(NDMET)
REAL*8             ECF(NDMET,NDCONF),        ECTAA(NDMET,NDCONF)
REAL*8             EDISGP,       EIONA(NDMET,2,NDORB)
REAL*8             ENPAR(NDMET),             FCF(NDMET,NDCONF)
REAL*8             FM(NDMET),    FM0(NDMET)
REAL*8             PARMD(NDMET,10,NDCONF),   PARMR(NDMET,2,4)
REAL*8             PCF(NDMET,NDCONF)
REAL*8             QDORB((NDQDN*(NDQDN+1))/2)
REAL*8             SAO(NDMET,2,NDTHET),      SCALGP
REAL*8             SPNFAC(NDMET,2),          THETA(NDTHET)
REAL*8             VORB(NDORB), W(NDMET),    WA(NDLEV)
REAL*8             WCF(NDMET,NDCONF),        WPA(NDLEV)
REAL*8             WVMAX(NDMET,NDCONF),      WVMIN(NDMET,NDCONF)
REAL*8             XITRUE(NDMET,2),          XJA(NDLEV)
REAL*8             XPJA(NDLEV)
```

5.44 d7grps: Subroutine d7grps from library adas4xx

```
C
       subroutine d7grps(nwave, ndwave, wave, index)


C-----------------------------------------------------------------------
C
C  ******************** FORTRAN77 FUNCTION: D7GRPS *********************
C
C  PURPOSE  : Puts wavelengths into a 5 bin histogram for automating
C             effective line selection in ADAS407.
C
C  COMMENTS : Note that the number of bins is fixed and no attempt is
C             made at clever adaptive methods of optimising case where
C             there is a large empty wavelength space between groups.
C
C  INPUT:  (I*4)   NWAVE   = number of wavelengths
C  INPUT:  (I*4)   NDWAVE  = max size of wavelength array
C  INPUT:  (R*8)   WAVE    = wavelengths
C
C  OUTPUT: (I*4)   INDEX   = indices (1->nbin) of wavelengths
C
C  ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4INDF     ADAS      Finds closet index in an array
C
C
C  AUTHOR   : Martin O'Mullane
C
C  VERSION  : 1.1
C  DATE     : 17-01-2002
C
C  MODIFIED : Martin O'Mullane
C              - First version.
C
C-----------------------------------------------------------------------
      INTEGER             INDEX(NDWAVE),            NDWAVE,      NWAVE
      REAL*8              WAVE(NDWAVE)
```

5.45 d7link: Subroutine d7link from library adas4xx

```
C UNIX-IDL PORT - SCCS INFO: MODULE @(#)d7link.for 1.2 DATE 02/27/98

      SUBROUTINE D7LINK( NDLEV   , NDMET  ,
     &                   NMET    , IMETR  , NPMET  , IPMETR ,
     &                   CSTRGA  , ISA    , ILA    , NALCM  , IALCM ,
     &                   ISALCM  ,
     &                   CSTRGPA , IPSA   , IPLA   , NALCP  , IALCP ,
     &                   ISALCP  ,
     &                   LLINK   , ILINK  , LEISS
     &                 )
C-----------------------------------------------------------------------
C                                                                       ********
C  ****************** FORTRAN 77 SUBROUTINE: D7LINK ***************************
C
C  PURPOSE: RETURNS A TRUTH TABLE OF LINKS BETWEEN PARENTS AND
C           RECOMBINED ION METASTABLES FOR RADIATIVE RECOMBINATION
C           AND IONISATION.  ALSO SUPPLIES THE DECIMAL ORBITAL NUMBER
C           FOR THE POSITION OF THE SHELL OF THE RECOMBINED ELECTRON.
C
C  CALLING PROGRAM: ADAS407
C
C  SUBROUTINE:
C
C  INPUT  :  (I*4)   NDLEV    = MAX. NUMBER OF LEVELS ALLOWED
C  INPUT  :  (I*4)   NDMET    = MAX. NO. OF METASTABLES ALLOWED
C  INPUT  :  (I*4)   NMET     = NUMBER OF METASTABLES (1<=NMET<=NDMET)
C  INPUT  :  (I*4)   IMETR()  = INDEX OF METASTABLE IN COMPLETE LEVEL
C                               LIST (ARRAY SIZE = 'NDMET' )
C  INPUT  :  (I*4)   NPMET    = NUMBER OF PARENT METASTABLES
C                               (1<=NPMET<=NDMET)
C  INPUT  :  (I*4)   IPMETR() = INDEX OF PARENT METASTABLES IN LEVEL
C                               LIST (ARRAY SIZE = 'NDMET' )
C  INPUT  :  (C*18)  CSTRGA() = CONFIGURATION (EISSNER FORM) FOR
C                               RECOMBINED ION LEVELS
C  INPUT  :  (I*4)   ILA()    = QUANTUM NUMBER (L) FOR LEVELS
C                               (RECOMNBINED  ION COPASE FILE)
C  INPUT  :  (I*4)   ISA()    = MULTIPLICITY FOR LEVELS
C                               (RECOMBINED ION COPASE FILE)
C                               NOTE: (ISA-1)/2 = QUANTUM NUMBER (S)
C  INPUT  :  (C*18)  CSTRGPA()= CONFIGURATION (EISSNER FORM) FOR
C                               RECOMBINING ION LEVELS
C  INPUT  :  (I*4)   IPLA()   = QUANTUM NUMBER (L) FOR LEVELS
C                               (RECOMBINING ION COPASE FILE)
C  INPUT  :  (I*4)   IPSA()   = MULTIPLICITY FOR LEVEL 'IA2()'
C                               (RECOMBINING ION COPASE FILE)
C                               NOTE: (IPSA-1)/2 = QUANTUM NUMBER (S)
C
C  OUTPUT :  (I*4)   NALCM    = NUMBER OF SPIN DISTINGUISED
C                               METASTABLES
C  OUTPUT :  (I*4)   IALCM()  = INDEX OF ENERGY ORDERED SPIN
C                               DISTINQUISHED METASTABLE
C                               1ST. DIM: METASTABLE INDEX
C  OUTPUT :  (I*4)   ISALCM() = SPIN OF ENERGY ORDERED SPIN
C                               DISTINQUISHED METASTABLE
C                               1ST. DIM: DISTINQUISHED METASTABLE INDEX
C  OUTPUT :  (I*4)   NALCP      NUMBER OF SPIN DISTINQUISHED
C                               PARENTS
C  OUTPUT :  (I*4)   IALCP()  = INDEX FOR ENERGY ORDERED SPIN
C                               DISTINQUISHED PARENT
```

```
C                                1ST. DIM: PARENT INDEX
C  OUTPUT :  (I*4)   ISALCP() = SPIN OF ENERGY ORDERED SPIN
C                                DISTINQUISHED PARENT
C                                1ST. DIM: DISTINQUISHED PARENT INDEX
C  OUTPUT :  (L*4)   LLINK(,,)= .TRUE.  => LINK EXISTS
C                                .FALSE. => NO LINK EXISTS
C                                 1ST DIM: METASTABLE INDEX
C                                 2ND DIM: PARENT METASTABLE INDEX
C                                 3RD DIM: SPEN SYSTEM INDEX
C  OUTPUT :  (L*4)   ILINK(,,)= DECIMAL ORBITAL INDEX FOR RECOMBINED
C                                ION ORBITAL DIFFERENCE WITH PARENT
C                                 1ST DIM: METASTABLE INDEX
C                                 2ND DIM: PARENT METASTABLE INDEX
C                                 3RD DIM: SPEN SYSTEM INDEX
C  OUTPUT :  (L*4)   LEISS    = .TRUE.  => ALL CONFIGS. EISSNER FORM
C                                .FALSE. => NOT ALL CONFIGS. EISSNER
C
C            (I*4)   NOCCUM() = OCCUPANCY FOR EACH DECIMAL ORBITAL
C                                INDEX 1-15 OF METASTABLE
C            (I*4)   NOCCUP() = OCCUPANCY FOR EACH DECIMAL ORBITAL
C                                INDEX 1-15 OF PARENT
C
C            (I*4)   I      = GENERAL INDEX
C            (I*4)   J      = GENERAL INDEX
C            (I*4)   IM     = GENERAL INDEX
C            (I*4)   IPAR   = GENERAL INDEX
C            (I*4)   IORBIT = CURRENT ORBITAL INDEX
C            (L*4)   LMATCH = GENERAL LOGICAL VARIABLE
C            (L*4)   LTYPE    = .TRUE.  => CONFIG. EISSNER FORM
C                               .FALSE. => CONFIG. NOT EISSNER FORM
C
C  ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          DXEXCF     ADAS      EXPAND EISSNER CONFIG. INTO SHELL OCCUP.
C          DXCOMP     ADAS      COMPARE TWO OCCUPANCY VECTORS
C          I4UNIT     ADAS      FETCH UNIT NUMBER FOR OUTPUT OF MESSAGES
C
C  AUTHOR:  H. P. SUMMERS, UNIVERSITY OF STRATHCLYDE
C           JA8.08
C           TEL. 0141-553-4196
C
C  DATE:    05/06/96
C
C  UPDATE:  24/07/96 - PEB - ADDED THIRD 'LTYPE' ARGUMENT TO 3RD AND 4TH
C                            CALLS TO ROUTINE DXEXCF. (IT HAD BEEN LEFT
C                            OFF.)
C
C UNIX-IDL PORT:
C    WILLIAM OSBORN, TESSELLA SUPPORT SERVICES PLC.
C
C DATE:    20TH AUGUST 1996
C
C VERSION: 1.1 DATE: 20-08-96
C MODIFIED: WILLIAM OSBORN
C     - FIRST VERSION
C
C VERSION: 1.2 DATE: 14-08-97
C MODIFIED: HUGH SUMMERS
C     - ADDED SPIN DISTINQUISHED PARENT AND METASTABLE
C             IDENTIFICATION, COUNTERS AND POINTERS
```

```
C
C VERSION: 1.3 DATE: 22-11-2003
C MODIFIED: Martin O'Mullane
C     - Pass configurations through ceprep before acting on them.
C     - Extend dimensions of orbital arrays.
C
C-----------------------------------------------------------------------
      CHARACTER*18        CSTRGA(NDLEV),            CSTRGPA(NDLEV)
      INTEGER             IALCM(NDMET),             IALCP(NDMET)
      INTEGER             ILA(NDLEV),  ILINK(NDMET,NDMET,2)
      INTEGER             IMETR(NDMET),             IPLA(NDLEV)
      INTEGER             IPMETR(NDMET),            IPSA(NDLEV)
      INTEGER             ISA(NDLEV),  ISALCM(NDMET)
      INTEGER             ISALCP(NDMET),            NALCM,       NALCP
      INTEGER             NDLEV,       NDMET,       NMET,        NPMET
      LOGICAL             LEISS,       LLINK(NDMET,NDMET,2)
```

5.46 d7lotz: Subroutine d7lotz from library adas4xx

```
      SUBROUTINE D7LOTZ(IZ0 ,  IZ1,
     &                  NPAR,
     &                  A   ,  B  ,  C)

C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: D7LOTZ *********************
C
C
C  PURPOSE:
C           (1) RETURNS LOTZ PARAMETERS FOR A GIVEN Z AND ION STAGE
C
C               W LOTZ, IPP 1/62 & IPP 1/76
C
C           FOR ELEMENTS AND IONISATION STAGES NOT INCLUDED RETURNS
C
C           A(1)=4.5, B(1)=0, C(1)=0     NPAR=2
C           A(2)=4.5, B(2)=0, C(2)=0
C
C           FOR H- AND HE- LIKE
C           A(1)=4.5, B(1)=0, C(1)=0     NPAR=1
C
C           NOTE: NO TRAP FOR IZ1>IZ
C
C
C
C  CALLING PROGRAM: ADAS407
C
C  SUBROUTINE:
C
C
C  INPUT   : (I*4)  IZ0      = NUCLEAR CHARGE
C  INPUT   : (I*4)  IZ1      = FINAL ION CHARGE
C
C
C  OUTPUT  : (R*8)  NPAR     = NUMBER OF PARAMETER GROUPS
C            (R*8)  A()      = LOTZ A PARAMETER
C                              1ST DIM: GROUP INDEX
C            (R*8)  B()      = LOTZ B PARAMETER
C                              1ST DIM: GROUP INDEX
C            (R*8)  C()      = LOTZ C PARAMETER
C                              1ST DIM: GROUP INDEX
C
C  PROGRAM
C          : (I*4)  IZDIMD   = NUMBER OF ELEMENTS OF LOTZ DATA
C          : (I*4)  ISDIMD   = NUMBER OF ION STAGES
C          : (I*4)  IPDIMD   = NUMBER OF A,B,C PARAMETERS
C
C          : (L*4)  LZFND    = .TRUE. ELEMENT HAS LOTZ PARAMETERS
C          : (L*4)  LSFND    = .TRUE. ION STAGE HAS LOTZ PARAMETERS
C
C
C NOTE:
C
C ROUTINES:   NONE
C
C
C
```

```
C AUTHOR:  M O'MULLANE
C
C DATE:    7/07/94
C
C UNIX-IDL PORT:
C    WILLIAM OSBORN, TESSELLA SUPPORT SERVICES PLC.
C
C DATE:    22ND APRIL 1996
C
C VERSION: 1.1 DATE: 22-04-96
C MODIFIED: WILLIAM OSBORN
C     - FIRST VERSION. NO CHANGES TO IBM CODE.
C-----------------------------------------------------------------------
      INTEGER             IZ0,          IZ1,          NPAR
      REAL*8              A(IPDIMD),    B(IPDIMD),    C(IPDIMD)
```

5.47 d7pyvr: Subroutine d7pyvr from library adas4xx

```
      SUBROUTINE D7PYVR(Y,Z1,PY)
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: D7PYVR *********************
C
C  PURPOSE : CALCULATES VAN REGEMORTER'S P FACTOR FOR ELECTRON
C            COLLISIONS WITH ATOMS AND IONS.
C
C  INPUT   : Y=ATE*(1/V1**2+1/V2**2) WITH
C            ATE - 1.5789D5*Z1**2/TE
C            TE  - ELECTRON TEMPERATURE (K)
C            V1  - INITIAL EFFECTIVE PRINCIPAL QUANTUM NUMBER
C            V2  - FINAL EFFECTIVE PRINCIPAL QUANTUM NUMBER
C            Z1  - TARGET ION CHARGE +1
C
C  OUTPUT  : PY  - P-FACTOR
C
C
C  AUTHOR  : H.P. SUMMERS, JET
C            K1/1/57
C            JET EXT. 4941
C
C  DATE:     03/06/94
C
C UNIX-IDL PORT:
C    WILLIAM OSBORN, TESSELLA SUPPORT SERVICES PLC.
C
C DATE:    22ND APRIL 1996
C
C VERSION: 1.1 DATE: 22-04-96
C MODIFIED: WILLIAM OSBORN
C     - FIRST VERSION. NO CHANGES TO IBM CODE.
C
C-----------------------------------------------------------------------
      REAL*8              PY,          Y,           Z1
```

5.48 d7spln: Subroutine d7spln from library adas4xx

```
      SUBROUTINE D7SPLN(         LOSEL ,
     &                   NV    , MAXT  , NPSPL  ,
     &                   SCEF  , TOA   , TOSA   ,
     &                   PTOT  , POWOA , POWOSA ,
     &                   LTRNG
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: D7SPLN *********************
C
C  (IDENTICAL TO: E6SPLN (EXCEPT SOME VARIABLE NAMES ARE CHANGED))
C
C  PURPOSE:
C         1) PERFORMS CUBIC SPLINE ON LOG(TEMPERATURE) VERSUS LOG(POW)
C            INPUT DATA. ('SCEF' VERSUS 'PTOT' , NV DATA PAIRS)
C
C         2) INTERPOLATES 'MAXT'  POW VALUES USING ABOVE SPLINES AT
C            TEMPERATURES READ IN FROM ISPF PANELS FOR TABULAR OUTPUT.
C            (ANY TEMPERATURE VALUES WHICH REQUIRED EXTRAPOLATION TO
C             TAKE PLACE ARE SET TO ZERO).
C                 - THIS STEP ONLY TAKES PLACE IF 'LOSEL=.TRUE.' -
C
C         3) INTERPOLATES 'NPSPL' POW VALUES USING ABOVE SPLINES AT
C            TEMPERATURES EQUI-DISTANCE ON RANGE OF LOG(TEMPERATURES)
C            STORED IN INPUT 'SCEF' ARRAY.
C
C  CALLING PROGRAM: ADAS407
C
C
C  SUBROUTINE:
C
C  INPUT : (L*4)  LOSEL   = .TRUE.  => CALCULATE POWS FOR INPUT TEMPS.
C                                      READ FROM ISPF PANEL.
C                           .FALSE. => - DO NOT DO THE ABOVE -
C
C  INPUT : (I*4)  NV      = INPUT DATA FILE: NUMBER OF POW/TEMPERATURE
C                           PAIRS READ FOR THE TRANSITION BEING ASSESSED
C  INPUT : (I*4)  MAXT    = NUMBER OF ISPF ENTERED TEMPERATURE VALUES AT
C                           WHICH INTERPOLATED POW VALUES ARE REQUIRED
C                           FOR TABULAR OUTPUT.
C  INPUT : (I*4)  NPSPL   = NUMBER OF  SPLINE  INTERPOLATED  POW/TEMP.
C                           REQUIRED FOR GRAPHICAL DISPLAY.
C
C  INPUT : (I*4)  SCEF()  = INPUT DATA FILE: TEMPERATURES (KELVIN)
C  INPUT : (I*4)  TOA()   = ISPF PANEL ENTERED TEMPERATURES (KELVIN)
C  OUTPUT: (I*4)  TOSA()  = 'NPSPL' TEMPERATURES FOR GRAPHICAL OUTPUT
C                           (KELVIN).
C
C  INPUT : (R*8)  PTOT()  = INPUT DATA FILE: SELECTED TRANSITION -
C                           POW VALUES AT 'SCEF()'.
C  OUTPUT: (I*4)  POWOA() = SPLINE INTERPOLATED POW VALUES AT 'TOA()'
C                           (EXTRAPOLATED VALUES = 0.0).
C  OUTPUT: (R*8)  POWOSA()= SPLINE INTERPOLATED POW VALUES AT 'TOSA()'
C
C  OUTPUT: (L*4)  LTRNG() = .TRUE.  => OUTPUT   SPLINE    VALUE    WAS
C                                      INTERPOLATED FOR 'DLOG(TOA()'.
C                           .FALSE. => OUTPUT   SPLINE    VALUE    WAS
C                                      EXTRAPOLATED FOR 'DLOG(TOA()'.
```

```
C                                     (NOTE: 'YOUT()=0' AS 'IOPT < 0').
C
C          (I*4)  NIN     = PARAMETER = MAX. NO. OF  INPUT  TEMP/POW
C                                       PAIRS MUST BE >= 'NV'
C          (I*4)  NOUT    = PARAMETER = MAX. NO. OF 'OUTPUT TEMP/POW
C                                       PAIRS MUST BE >= 'MAXT' & 'NPSPL'
C
C          (I*4)  IARR    = ARRAY SUBSCRIPT USED FOR TEMP/POW PAIRS
C          (I*4)  IOPT    = DEFINES THE BOUNDARY DERIVATIVES FOR THE
C                           SPLINE ROUTINE 'XXSPLE', SEE 'XXSPLE'.
C                           (VALID VALUES = <0, 0, 1, 2, 3, 4)
C
C          (R*8)  TSTEP   = THE SIZE OF STEP BETWEEN 'XOUT()' VALUES FOR
C                           GRAPHICAL  OUTPUT  TEMP/POW  PAIRS  TO  BE
C                           CALCULATED USING SPLINES.
C
C          (L*4)  LSETX   = .TRUE.  => SET UP SPLINE PARAMETERS RELATING
C                                      TO 'XIN' AXIS.
C                           .FALSE. => DO NOT SET UP SPLINE PARAMETERS
C                                      RELATING TO 'XIN' AXIS.
C                                      (I.E. THEY WERE SET IN A PREVIOUS
C                                            CALL )
C                           (VALUE SET TO .FALSE. BY 'XXSPLE')
C
C          (R*8)  XIN()   = LOG( 'SCEF()' )
C          (R*8)  YIN()   = LOG( 'PTOT()' )
C          (R*8)  XOUT()  = LOG(TEMPERATURES AT WHICH SPLINES REQUIRED)
C          (R*8)  YOUT()  = LOG(OUTPUT SPLINE INTERPOLATED POW VALUES)
C          (R*8)  DF()    = SPLINE INTERPOLATED DERIVATIVES
C
C          (L*4)  LDUMP() = .TRUE.  => OUTPUT SPLINE VALUE INTRPOLATED
C                                      FOR 'YOUT()'.
C                           .FALSE. => OUTPUT SPLINE VALUE EXTRAPOLATED
C                                      FOR 'YOUT()'.
C                                      (NOTE: USED AS A DUMMY ARGUMENT.
C                                             ALL VALUES WILL BE TRUE.)
C
C NOTE:
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          XXSPLE     ADAS      SPLINE SUBROUTINE (EXTENDED DIAGNOSTICS)
C          R8FUN1     ADAS      REAL*8 FUNCTION: ( X -> X )
C
C AUTHOR:  H. P. SUMMERS, JET
C          K1/1/57
C          JET EXT. 4941
C
C DATE:    10/06/94
C
C UNIX-IDL PORT:
C    WILLIAM OSBORN, TESSELLA SUPPORT SERVICES PLC.
C
C DATE:    22ND APRIL 1996
C
C VERSION: 1.1 DATE: 22-04-96
C MODIFIED: WILLIAM OSBORN
C     - FIRST VERSION. NO CHANGES TO IBM CODE
C-----------------------------------------------------------------------
C
```

```
C-----------------------------------------------------------------------
      INTEGER             MAXT,         NPSPL,        NV
      LOGICAL             LOSEL,        LTRNG(MAXT)
      REAL*8              POWOA(MAXT), POWOSA(NPSPL)
      REAL*8              PTOT(NV),     SCEF(NV),     TOA(MAXT)
      REAL*8              TOSA(NPSPL)
```

5.49 d7wr11: Subroutine d7wr11 from library adas4xx

```
      SUBROUTINE D7WR11( IUNIT  ,
     &                   NDMET  , NDRHO   , NDTHET  , NDREP   , NDCONF ,
     &                   NDORB  , NDQDN   ,
     &                   IZ0    , IZ      ,
     &                   NPMET  , NRHO    , NTHETA  , INREP   ,
     &                   ISPRT  , ISPSYS  , ITYPEA  , IZETA4  ,
     &                   NCUT   , NSYS    , NTRANS  , N0A     , NREP   ,
     &                   NCTAA  , NZETA   , N1A     ,
     &                   Z1     , EDISGP  , SCALGP  , CI4     ,
     &                   ALFRA  , ADIELO  , EIONA   , SAO     ,
     &                   THETA  , RHO     , RHOP    ,
     &                   PARMR  , PARMD   ,
     &                   WNREP  ,
     &                   TRMPRT ,
     &                   NALCM  , ISALCM  , NALCP   , ISALCP  ,
     &                   QDN
     &                 )
C-----------------------------------------------------------------------
C
C  ******************** FORTRAN77 SUBROUTINE: D7WR11 ******************
C
C  PURPOSE:  TO OUTPUT DATA TO MAINBN PASSING FILE.
C            DATA  FOR INITIATING A MAINBNS BUNDLE-NS CALCULATION
C
C  CALLING PROGRAM: ADAS407
C
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  IUNIT   = OUTPUT UNIT NUMBER FOR RESULTS
C  INPUT : (I*4)  IZ      = RECOMBINED ION CHARGE
C                           (1ST COPASE FILE)
C  INPUT : (I*4)  NDQDN   = MAX. NUMBER OF N-SHELLS FOR QUANTUM DEFECTS
C  INPUT : (I*4)  NALCM   = NUMBER OF SPIN DISTINGUISED
C                            METASTABLES
C  INPUT : (I*4)  ISALCM()= SPIN OF ENERGY ORDERED SPIN
C                           DISTINQUISHED METASTABLE
C                           1ST. DIM: DISTINGUISHED METASTABLE INDEX
C  INPUT : (I*4)  NALCP     NUMBER OF SPIN DISTINGUISHED
C                           PARENTS
C  INPUT : (I*4)  ISALCP()= SPIN OF ENERGY ORDERED SPIN
C                           DISTINGUISHED PARENT
C                           1ST. DIM: DISTINGUISHED PARENT INDEX
C  INPUT : (R*8)  QDN()   = QUANTUM DEFECT FOR N-SHELLS.  NON-ZERO ONLY
C                           FOR ADF04 FILES WITH ORBITAL ENERGY DATA
C                           1ST. DIM: N-SHELL (1<=N<=NDQDN)
C
C
C          (I*4) I         = GENERAL USE
C          (I*4) IFIRST    = GENERAL USE
C          (I*4) ILAST     = GENERAL USE
C
C NOTE:
C          THIS OUTPUT DATA IS FOR SUBSEQUENT INPUT TO A BACKGROUND
C          EXECUTION OF THE POPULATION  PROGRAM 'MAINBNS'.
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
```

```
C          XFESYM     ADAS      OBTAIN ELEMENT SYMBOL FROM NUCL. CHARGE
c          XXSLEN     ADAS      FIRST AND LAST NON-BLANK CHRS. OF STRNG
C
C AUTHOR:  H. P. SUMMERS, JET
C          K1/1/57
C          JET EXT. 4941
C
C DATE:    01/07/94
C
C UNIX-IDL PORT:
C    WILLIAM OSBORN, TESSELLA SUPPORT SERVICES PLC.
C
C DATE:    22ND APRIL 1996
C
C VERSION: 1.1 DATE: 22-04-96
C MODIFIED: WILLIAM OSBORN
C     - FIRST VERSION.
C
C VERSION: 1.2 DATE: 20-08-96
C MODIFIED: HUGH SUMMERS + WILLIAM OSBORN
C     - MADE THE REPRESENTATIVE LEVEL LIST
C             TO THE MAINBN FILE BEGIN AT 1.
C
C
C VERSION: 1.3 DATE: 14-08-97
C MODIFIED: HUGH SUMMERS
C     - ADDED SPIN DISTINQUISHED PARENT AND METASTABLE
C             IDENTIFICATION, COUNTERS AND POINTERS.  USING
C             IGRD TO MARK THE SPIN SYSTEM NOW.  PUT OUT
C             TRUE QUANTUM DEFECTS.  IZ INCLUDED IN CALL
C             VARAIBLES.
C
C VERSION: 1.4 DATE: 07-12-98
C MODIFIED: Martin O'Mullane
C     - Extended MAINBNS namelist to give the ionisation (adf07)
C             and DR cross reference file in the adf25 driver.
C
C-----------------------------------------------------------------------
      CHARACTER*2         TRMPRT(NDMET)
      INTEGER             INREP,       ISALCM(NDMET)
      INTEGER             ISALCP(NDMET),            ISPRT(NDMET)
      INTEGER             ISPSYS(NDMET,2),          ITYPEA(NDMET,NDCONF)
      INTEGER             IUNIT,       IZ,          IZ0
      INTEGER             IZETA4(NDMET,2,NDORB),    N0A(NDMET,2)
      INTEGER             N1A(NDMET,NDCONF),        NALCM,       NALCP
      INTEGER             NCTAA(NDMET,NDCONF),      NCUT(NDMET), NDCONF
      INTEGER             NDMET,       NDORB,       NDQDN,       NDREP
      INTEGER             NDRHO,       NDTHET,      NPMET
      INTEGER             NREP(NDREP), NRHO,        NSYS(NDMET), NTHETA
      INTEGER             NTRANS(NDMET),            NZETA(NDMET,2)
      REAL*8              ADIELO(NDMET,2,NDTHET)
      REAL*8              ALFRA(NDMET,2,NDTHET),    CI4,         EDISGP
      REAL*8              EIONA(NDMET,2,NDORB)
      REAL*8              PARMD(NDMET,10,NDCONF),   PARMR(NDMET,2,4)
      REAL*8              QDN(NDQDN),  RHO(NDRHO),  RHOP(NDRHO)
      REAL*8              SAO(NDMET,2,NDTHET),      SCALGP
      REAL*8              THETA(NDTHET),            WNREP(NDREP)
      REAL*8              Z1
```

5.50 d7wr12: Subroutine d7wr12 from library adas4xx

```
      SUBROUTINE D7WR12( IUNIT  , LWR12   ,
     &                   NDMET  , NDCONF  , NDORB   , NDBNDL  ,
     &                   CSELR  , CSELD   , CSELS   , CSELP   , CSELL  ,
     &                   IZ0    , IZ      , ISG     ,
     &                   ISPSYS , ITYPEA  , IZETA4  ,
     &                   NSYS   , NTRANS  , N0A     ,
     &                   NCTAA  , NZETA   , N1A     ,
     &                   Z1     , CI4     , EIONA   ,
     &                   PARMR  , PARMD   ,
     &                   IBNDL  ,
     &                   DEBNDL , FBNDL   , GBNDL   , PNLSA   ,
     &                   WVSPEC , IFSPEC  ,
     &                   DESPEC , FSPEC   , GSPEC   , SNLSA
     &                 )
C-----------------------------------------------------------------------
C
C  ******************** FORTRAN77 SUBROUTINE: D7WR12 ******************
C
C  PURPOSE:  TO OUTPUT DATA TO ATOMPARS PASSING FILE.
C            DATA  FOR INITIATING AN ADAS408 DATA PREPARATION RUN
C
C  CALLING PROGRAM: ADAS407
C
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  IUNIT   = OUTPUT UNIT NUMBER FOR RESULTS
C
C
C          (I*4) I         = GENERAL USE
C          (I*4) ISG       = MULTIPLICITY OF GROUND STATE OF RECOMBINED
C                            ION
C          (I*4) ITYPDN()  = SPECIFIES DELTA N FOR TRANSITION TYPE (1-7)
C          (I*4) ITYPMZ()  = SPECIFIES MERTZ CORRECTION ON(1) OR OFF(0)
C                            FOR TRANSITION TYPE (1-7)
C
C NOTE:
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          D7LOTZ     ADAS      RETURNS THE LOTZ IONISATION PARAMETERS
C
C AUTHOR:  H. P. SUMMERS, JET
C          K1/1/57
C          JET EXT. 4941
C
C DATE:    01/07/94
C
C UNIX-IDL PORT:
C    WILLIAM OSBORN, TESSELLA SUPPORT SERVICES PLC.
C
C DATE:    22ND APRIL 1996
C
C VERSION: 1.1 DATE: 22-04-96
C MODIFIED: WILLIAM OSBORN
C     - FIRST VERSION.
C
```

```
C VERSION: 1.2                          DATE: 13-05-96
C MODIFIED: TIM HAMMOND, TESSELLA SUPPORT SERVICES PLC.
C           - COMMENTED OUT REFERENCES TO VARIABLE IZO ('oh', not 'zero')
C
C VERSION: 1.3                                  DATE: 23-05-2003
C MODIFIED: Martin O'Mullane
C - Remove all unused variables and reduced length of
C                 parameter list.
C               - In cases where there is no spin system connection
C                 between adjacent ions force it to be ground. The
C                 proper solution is to use IC and not LS input files.
C
C-----------------------------------------------------------------------
      CHARACTER           CSELD,       CSELL,       CSELP,       CSELR
      CHARACTER           CSELS
      INTEGER             IBNDL,       IFSPEC(NDMET),            ISG
      INTEGER             ISPSYS(NDMET,2),          ITYPEA(NDMET,NDCONF)
      INTEGER             IUNIT,       IZ,          IZ0
      INTEGER             IZETA4(NDMET,2,NDORB),    N0A(NDMET,2)
      INTEGER             N1A(NDMET,NDCONF),        NCTAA(NDMET,NDCONF)
      INTEGER             NDBNDL,      NDCONF,      NDMET,       NDORB
      INTEGER             NSYS(NDMET), NTRANS(NDMET)
      INTEGER             NZETA(NDMET,2)
      LOGICAL             LWR12
      REAL*8              CI4,         DEBNDL(NDBNDL,NDMET)
      REAL*8              DESPEC(NDMET),            EIONA(NDMET,2,NDORB)
      REAL*8              FBNDL(NDBNDL,NDMET),      FSPEC(NDMET)
      REAL*8              GBNDL(NDBNDL,NDMET),      GSPEC(NDMET)
      REAL*8              PARMD(NDMET,10,NDCONF),   PARMR(NDMET,2,4)
      REAL*8              PNLSA(NDBNDL,NDMET),      SNLSA(NDMET)
      REAL*8              WVSPEC(NDMET),            Z1
```

5.51 d8data: Subroutine d8data from library adas4xx

```
       SUBROUTINE D8DATA( IUNIT,   IZDIMD,  IGDIMD,
     +                    IZ0,     IZL,     IZU,
     +                    IZRA,    IZDA,    IZIA,    IZTA,    IZSA,
     +                    CRRCA,   NRRCA,   ISRRCA,
     +                    NZA,     KSIA,
     +                    N0RA,    V0RA,    PHFCRA,  EDSPRA,  SCLERA,
     +                    CDRCA,   NDRCA,   ISDRCA,
     +                    DEDA,    FDA,     GDA,     NNDA,    MSDA,
     +                    ITYPDA,  N0DA,    NCUTA,   V0DA,    PHFCDA,
     +                    CRFCDA,  EPSIJA,  FIJA,    EDSPDA,  SCLEDA,
     +                    CCIOA,   NCIOSA,  NCIORA,  ISCIOA,
     +                    PIOA,    AIOA,    BIOA,    CIOA,    NQIOA,
     +                    ZETAA,   EIONA,   CIA,
     +                    WGHTA,   ENERA,   CRA,
     +                    CPLTA,   NPLTA,   ISPLTA,
     +                    DEPTA,   FPTA,    GPTA,    NNPTA,   SPYLTA,
     +                    CPLSA,   NPLSA,   ISPLSA,  INFO,
     +                    DEPSA,   FPSA,    GPSA,    NNPSA,   SPYLSA,
     +                    LVALID
     +                  )
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: D8DATA *********************
C
C  PURPOSE:  TO FETCH DATA FROM INPUT ATOMPARS DATA SET OF TYPE ADF03.
C
C  CALLING PROGRAM: ADAS408
C
C
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  IUNIT     = UNIT TO WHICH INPUT FILE IS ALLOCATED
C  INPUT : (I*4)  IZDIMD    = MAXIMUM NUMBER OF IONISATION STAGES
C  INPUT : (I*4)  IGDIMD    = MAXIMUM NUMBER OF GROUPS
C
C
C  THE OUTPUT ARRAYS ARE INDEXED
C            XXX()  = XXX(IZDIMD)         1ST DIMENSION ION STAGE
C
C            XXX(,) = XXX(IZDIMD,IGDIMD)  1ST DIMENSION ION STAGE
C                                         2ND DIMENSION GROUP
C
C
C
C  OUTPUT: (I*4)  IZ0       =  NUCLEAR CHARGE
C  OUTPUT: (I*4)  IZL       =  LOWEST INCLUDED ION
C  OUTPUT: (I*4)  IZU       =  HIGHEST INCLUDED ION
C
C  OUTPUT: (I*4)  IZRA()    =  RECOMBINING ION (RAD.  RECOM.)
C  OUTPUT: (I*4)  IZDA()    =  RECOMBINING ION (DIEL. RECOM.)
C  OUTPUT: (I*4)  IZIA()    =  IONISING ION    (COLL. IONIS.)
C  OUTPUT: (I*4)  IZTA()    =  RADIATING ION   (TOTAL LINE POWER)
C  OUTPUT: (I*4)  IZSA()    =  RADIATING ION   (SPECIFIC LINE POWER)
C
C
C
C  OUTPUT: (C*5)  CRRCA()   =  RADIATIVE RECOM. CODE
```

```
C  OUTPUT: (I*4)  NRRCA()    =    - NOT USED -
C  OUTPUT: (I*4)  ISRRCA()   =    - NOT USED -
C
C  OUTPUT: (I*4)  NZA()      = LOWEST ACCESSIBLE SHELL FOR RAD. RECOM.
C  OUTPUT: (I*4)  KSIA()     = NUMBER OF ELECTRONS IN SHELL
C
C  OUTPUT: (I*4)  N0RA()     = LOWEST ACCESSIBLE PRINC. QUANTUM SHELL
C                              FOR RAD. RECOM.
C  OUTPUT: (I*4)  V0RA()     = EFFECTIVE PRINCIPAL QUANTUM NUMBER
C                              FOR SHELL
C  OUTPUT: (R*8)  PHFCRA()   = PHASE SPACE OCCUPANCY AVAILABILITY
C                              FOR SHELL
C  OUTPUT: (R*8)  EDSPRA()   = ENERGY ADJUSTMENT IN LOWEST SHELL
C                              RATE COEFFICIENT
C  OUTPUT: (R*8)  SCLERA()   = MULTIPLIER FOR LOWEST SHELL
C                              RATE COEFFICIENT
C
C
C
C  OUTPUT: (C*5)  CDRCA()    =  DIELECTRONIC RECOM. CODE
C  OUTPUT: (I*4)  NDRCA()    =  NUMBER OF TRANSITIONS FOLLOWING
C  OUTPUT: (I*4)  ISDRCA()   =    - NOT USED -
C
C  OUTPUT: (R*8)  DEDA(,)    = TRANSITION ENERGY (EV)
C  OUTPUT: (R*8)  FDA(,)     = OSCILLATOR STRENGTH
C  OUTPUT: (R*8)  GDA(,)     = GAUNT FACTOR
C  OUTPUT: (I*4)  NNDA(,)    = DELTA N FOR TRANSITION
C  OUTPUT: (I*4)  MSDA(,)    = MERTZ SWITCH (0=OFF, 1=ON)
C
C  OUTPUT: (I*4)  ITYPDA(,)  = TYPE OF DIELECTRONIC TRANSITION
C  OUTPUT: (I*4)  N0DA(,)    = LOWEST ACCESSIBLE PRINC. QUANTUM SHELL
C                              FOR DIEL. RE
C  OUTPUT: (I*4)  NCUTA(,)   = CUT-OFF PRINC. QUANTUM SHELL IN
C                              GENERAL PROGRAM
C  OUTPUT: (I*4)  V0DA(,)    = EFFECTIVE PRINC. QUANTUM NUMBER
C                              FOR LOWEST ACCESS
C  OUTPUT: (R*8)  PHFCDA(,)  = PHASE SPACE OCCUPANCY AVAILABILITY
C                              FOR LOWEST SHELL
C  OUTPUT: (R*8)  CRFCDA(,)  = ADJUSTMENT FOR BETHE CORRECTIONS
C                              IN GENERAL PROGRAM
C  OUTPUT: (R*8)  EPSIJA(,)  = Z-SCALED PARENT TRANSITION ENERGY (RYD)
C  OUTPUT: (R*8)  FIJA(,)    = OSCILLATOR STRENGTH FOR TRANSITION
C  OUTPUT: (R*8)  EDSPDA(,)  = ENERGY ADJUSTMENT IN BURGESS GENERAL
C                              FORMULA (RYD)
C  OUTPUT: (R*8)  SCLEDA(,)  = MULTIPLIER ON BURGESS GENERAL FORMULA
C
C
C
C  OUTPUT: (C*5)  CCIOA()    =  COLLISIONAL IONIS. CODE
C  OUTPUT: (I*4)  NCIOSA()   =  NUMBER OF SHELL  VALUES FOLLOWING
C  OUTPUT: (I*4)  NCIORA()   =  NUMBER OF RESON. VALUES FOLLOWING
C  OUTPUT: (I*4)  ISCIOA()   =    - NOT USED -
C
C  OUTPUT: (R*8)  PIOA(,)    = SHELL IONISATION POTENTIAL (EV)
C  OUTPUT: (R*8)  AIOA(,)    = LOTZ PARAMETER
C  OUTPUT: (R*8)  BIOA(,)    = LOTZ PARAMETER
C  OUTPUT: (R*8)  CIOA(,)    = LOTZ PARAMETER
C  OUTPUT: (I*4)  NQIOA(,)   = EQUIVALENT ELECTRONS IN SHELL
C
C  OUTPUT: (R*8)  ZETAA(,)   = NUMBER OF EQUIVALENT ELECTRONS FOR SHELL
C  OUTPUT: (R*8)  EIONA(,)   = IONISATION ENERGY FOR SHELL (RYD)
```

```
C  OUTPUT: (R*8)  CIA(,)    = MULTIPLIER FOR BURGESS-CHIDICHIMO RATE
C                             FOR SHELL
C  OUTPUT: (R*8)  WGHTA(,)  = WEIGHTING FACTOR FOR EXCITATION TO
C                             RESONANCE
C  OUTPUT: (R*8)  ENERA(,)  = EXCITATION ENERGY FOR TRANSITION
C                             TO RESONANCE (RYD)
C  OUTPUT: (R*8)  CRA(,)    = MULTIPLIER ON EXCITATION RATE EXPRESSSION
C
C
C
C  OUTPUT: (C*5)  CPLTA()   =  TOTAL LINE POWER CODE
C  OUTPUT: (I*4)  NPLTA()   =  NUMBER OF TRANSITIONS FOLLOWING
C  OUTPUT: (I*4)  ISPLTA()  =    - NOT USED -
C
C  OUTPUT: (R*8)  DEPTA(,)  = TRANSITION ENERGY (EV)
C  OUTPUT: (R*8)  FPTA(,)   = OSCILLATOR STRENGTH
C  OUTPUT: (R*8)  GPTA(,)   = GAUNT FACTOR
C  OUTPUT: (I*4)  NNPTA(,)  = DELTA N FOR TRANSITION
C
C  OUTPUT: (R*8)  SPYLTA(,) = MULTIPLIER OF VAN REGEMORTER P
C                             FACTOR IN TOTAL POWER
C
C
C
C  OUTPUT: (C*5)  CPLSA()   = SPECIFIC LINE POWER CODE
C  OUTPUT: (I*4)  NPLSA()   =   - NOT USED -
C  OUTPUT: (I*4)  ISPLSA()  =   - NOT USED -
C  OUTPUT: (C*8)  INFO()    = WAVELENGTH OF SPECIFIC LINE FOR
C                             NAMING PURPOSES
C
C  OUTPUT: (R*8)  DEPSA(,)  = TRANSITION ENERGY (EV)
C  OUTPUT: (R*8)  FPSA(,)   = OSCILLATOR STRENGTH
C  OUTPUT: (R*8)  GPSA(,)   = GAUNT FACTOR
C  OUTPUT: (I*4)  NNPSA(,)  = DELTA N FOR TRANSITION
C
C  OUTPUT: (R*8)  SPYLSA(,) = MULTIPLIER OF VAN REGEMORTER P FACTOR
C                             IN SPECIFIC LINE POWER
C
C
C
C  OUTPUT: (L*4)  LVALID  = .TRUE. DATA SET READ AND APPEARS VALID
C                         = .FALSE. ERROR DETECTED IN READING DATA SET
C
C
C
C  PROGRAM:
C          (C*50) ERRMSG()= ERROR MESSAGE STRING
C          (I*4)  IDUM    = PROGRAM USE
C
C
C  ROUTINES:
C          ROUTINE   SOURCE   DESCRIPTION
C          --------------------------------------------------
C          I4UNIT    ADAS     FETCH UNIT NUMBER FOR MESSAGE OUTPUT
C
C
C  AUTHOR:  H. P. SUMMERS, JET
C          K1/1/57
C          JET EXT. 4941
C
C  DATE:    10/05/94
```

```
C
C  UPDATE:
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 04-04-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - PUT UNDER S.C.C.S. CONTROL
C
C VERSION: 1.2 DATE: 07-04-97
C MODIFIED: RICHARD MARTIN
C ADDED THE LINE WRITE(I4UNIT(-1),*)STRG1
C
C VERSION: 1.3 DATE: 24-02-98
C MODIFIED: M OMULLANE
C CHANGED 'E6DATA' TO 'D8DATA' IN FORMAT STATEMENTS 2001 & 2009
C
C-----------------------------------------------------------------------
C     GENERAL VARIABLES
C
      CHARACTER*5         CCIOA(IZDIMD),            CDRCA(IZDIMD)
      CHARACTER*5         CPLSA(IZDIMD),            CPLTA(IZDIMD)
      CHARACTER*5         CRRCA(IZDIMD)
      CHARACTER*8         INFO(IZDIMD)
      INTEGER             IGDIMD,      ISCIOA(IZDIMD)
      INTEGER             ISDRCA(IZDIMD),           ISPLSA(IZDIMD)
      INTEGER             ISPLTA(IZDIMD),           ISRRCA(IZDIMD)
      INTEGER             ITYPDA(IZDIMD,IGDIMD),    IUNIT,       IZ0
      INTEGER             IZDA(IZDIMD),             IZDIMD
      INTEGER             IZIA(IZDIMD),             IZL
      INTEGER             IZRA(IZDIMD),             IZSA(IZDIMD)
      INTEGER             IZTA(IZDIMD),             IZU
      INTEGER             KSIA(IZDIMD),             MSDA(IZDIMD,IGDIMD)
      INTEGER             N0DA(IZDIMD,IGDIMD),      N0RA(IZDIMD)
      INTEGER             NCIORA(IZDIMD),           NCIOSA(IZDIMD)
      INTEGER             NCUTA(IZDIMD,IGDIMD),     NDRCA(IZDIMD)
      INTEGER             NNDA(IZDIMD,IGDIMD),      NNPSA(IZDIMD,IGDIMD)
      INTEGER             NNPTA(IZDIMD,IGDIMD),     NPLSA(IZDIMD)
      INTEGER             NPLTA(IZDIMD),            NQIOA(IZDIMD,IGDIMD)
      INTEGER             NRRCA(IZDIMD),            NZA(IZDIMD)
      INTEGER             V0RA(IZDIMD)
      LOGICAL             LVALID
      REAL*8              AIOA(IZDIMD,IGDIMD),      BIOA(IZDIMD,IGDIMD)
      REAL*8              CIA(IZDIMD,IGDIMD),       CIOA(IZDIMD,IGDIMD)
      REAL*8              CRA(IZDIMD,IGDIMD)
      REAL*8              CRFCDA(IZDIMD,IGDIMD),    DEDA(IZDIMD,IGDIMD)
      REAL*8              DEPSA(IZDIMD,IGDIMD),     DEPTA(IZDIMD,IGDIMD)
      REAL*8              EDSPDA(IZDIMD,IGDIMD),    EDSPRA(IZDIMD)
      REAL*8              EIONA(IZDIMD,IGDIMD),     ENERA(IZDIMD,IGDIMD)
      REAL*8              EPSIJA(IZDIMD,IGDIMD),    FDA(IZDIMD,IGDIMD)
      REAL*8              FIJA(IZDIMD,IGDIMD),      FPSA(IZDIMD,IGDIMD)
      REAL*8              FPTA(IZDIMD,IGDIMD),      GDA(IZDIMD,IGDIMD)
      REAL*8              GPSA(IZDIMD,IGDIMD),      GPTA(IZDIMD,IGDIMD)
      REAL*8              PHFCDA(IZDIMD,IGDIMD),    PHFCRA(IZDIMD)
      REAL*8              PIOA(IZDIMD,IGDIMD)
      REAL*8              SCLEDA(IZDIMD,IGDIMD),    SCLERA(IZDIMD)
      REAL*8              SPYLSA(IZDIMD,IGDIMD)
      REAL*8              SPYLTA(IZDIMD,IGDIMD),    V0DA(IZDIMD,IGDIMD)
      REAL*8              WGHTA(IZDIMD,IGDIMD),     ZETAA(IZDIMD,IGDIMD)
```

5.52 d8eval: Subroutine d8eval from library adas4xx

```
       SUBROUTINE D8EVAL ( dsn03    , dsn35    ,
     &                     NUMTE    , TEMIN    , TEMAX    ,
     &                     NUMNE    , FNEMIN   , FNEMAX   ,
     &                     FUELMAS  ,
     &                     TEL      , FNEL     , NAME     ,
     &                     iz0      , izl      , izu      ,
     &                     RAL      , SAL0     , SAL      , CAL     ,
     &                     PRBL     , PRCL     , PLTL0    , PLTL    ,
     &                     PLSL0    , PLSL     ,
     &                     PRBLNFL  , PRCLNFL  , PLTL0NFL , PLTLNFL ,
     &                     ralrr    , raldr    ,
     &                     prbrr    , prbdr    , prbbr    ,
     &                     prbrrnfl , prbdrnfl , prbbrnfl ,
     &                     INFRAL   , INFSAL   , INFCAL   ,
     &                     INFPRB   , INFPRC   , INFPLT   , INFPLS  ,
     &                     INFPRBN  , INFPRCN  , INFPLTN  ,
     &                     METRAL   , METSAL   , METCAL   ,
     &                     METPRB   , METPRC   , METPLT   , METPLS  ,
     &                     METPRBN  , METPRCN  , METPLTN  ,
     &                     ltick
     &                   )



C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: D8EVAL *********************
C
C
C  PURPOSE:  To calculate tables of values of ionisation, recombination
C            and radiated power rates for non-coronal impurity studies
C            over a given temperature and density range with atomic
C            data parameters from files of format adf03
C
C  CALLING PROGRAM: ADAS408
C
C
C
C  SUBROUTINE:
C
C  INPUT:  (C*80) dsn03       = adf03 atompars file
C  INPUT:  (C*80) dsn35       = adf35 filter data file
C
C
C
C  THE OUTPUT ARRAYS ARE INDEXED
C          XXX(,) = XXX(ITDIMD,IDDIMD,IZDIMD) 1ST DIMENSION TEMPERATURE
C                                              2ND DIMENSION DENSITY
C                                              3RD DIMENSION ION STAGE
C  ALL TABLES ARE LOG10 IN CGS UNITS
C
C
C  OUTPUT: (R*8)  TEL()       = TEMPERATURE SET OF TABLES - LOG MESH
C          (R*8)  FNEL()      = DENSITY SET OF TABLES - LOG MESH
C          (C*13) NAME        = ELEMENT NAME
C
C          (R*8)  RAL(,,)     = RADIATIVE AND DIELECTRONIC RECOMB.
C          (R*8)  CAL(,,)     = CHARGE EXCHANGE RECOMBINATION
```

```
C
C          (R*8)  SAL0(,)      = IONISATION (NEUTRAL)
C          (R*8)  SAL(,,)      = IONISATION (NON-NEUTRAL)
C
C          (R*8)  PLTL0(,)     = TOTAL LINE RADIATED POWER (NEUTRAL)
C          (R*8)  PLTL(,,)     = TOTAL LINE RADIATED POWER (NON-NEUTRAL)
C
C          (R*8)  PRBL(,,)     = RAD. + DIEL RECOM. + BREMS. POWER
C          (R*8)  PRCL(,,)     = CX. RECOM. POWER
C
C          (R*8)  PLSL0(,)     = SPECIFIC LINE POWER   (NEUTRAL)
C          (R*8)  PLSL(,,)     = SPECIFIC LINE POWER   (NON-NEUTRAL)
C
C          (R*8)  PRBLNFL(,,) = RECOM+BREMM POWER (NO FILTER)
C          (R*8)  PRCLNFL(,,) = CX POWER (NO FILTER)
C          (R*8)  PLTLNFL(,,) = LINE POWER (NO FILTER)
C          (R*8)  PLTL0NFL(,,)= NEUTRAL LINE POWER (NO FILTER)
C
C          (C*8)  INFRAL()     = RECOMBINATION INFO STRING
C          (C*8)  INFSAL()     = IONISATION INFO STRING
C          (C*8)  INFCAL()     = CX INFO STRING
C          (C*8)  INFPRB()     = RECOM+BREMM POWER INFO STRING
C          (C*8)  INFPRC()     = CX POWER INFO STRING
C          (C*8)  INFPLT()     = TOTAL LINE POWER INFO STRING
C          (C*8)  INFPLS()     = SPECIFIC LINE POWER INFO STRING
C          (C*8)  INFPRBN()    = RECOMM+BREM POWER (NO FILTER) INFO
C          (C*8)  INFPRCN()    = CX POWER (NO FILTER) INFO STRING
C          (C*8)  INFPLTN()    = TOTAL LINE POWER (NO FILTER) INFO
C
C          (C*24) METRAL       = RECOMBINATION METHOD STRING
C          (C*24) METSAL       = IONISATION METHOD STRING
C          (C*24) METCAL       = CX METHOD STRING
C          (C*24) METPRB       = RECOM+BREMM POWER METHOD STRING
C          (C*24) METPRC       = CX POWER METHOD STRING
C          (C*24) METPLT       = TOTAL LINE POWER METHOD STRING
C          (C*24) METPLS       = SPECIFIC LINE POWER METHOD STRING
C          (C*24) METPRBN      = RECOMM+BREM POWER (NO FILTER) METHOD
C          (C*24) METPRCN      = CX POWER (NO FILTER) METHOD STRING
C          (C*24) METPLTN      = TOTAL LINE POWER (NO FILTER) METHOD
C
C
C  PROGRAM:
C
C  THE PARAMETER ARRAYS ARE INDEXED
C             XXX()  = XXX(IZDIMD)          1ST DIMENSION ION STAGE
C
C             XXX(,) = XXX(IZDIMD,IGDIMD)   1ST DIMENSION ION STAGE
C                                           2ND DIMENSION GROUP
C
C
C  INPUT:  (I*4)  IZDIMD     = MAXIMUM NUMBER OF IONISATION STAGES
C          (I*4)  IGDIMD     = MAXIMUM NUMBER OF GROUPS
C
C          (I*4)  ITDIMD     = MAXIMUM NUMBER OF TEMPERATURES
C          (I*4)  IDDIMD     = MAXIMUM NUMBER OF DENSITIES
C
C          (R*8)  TEMIN      = MINIMIUM TEMPERATURE OF TABLE
C          (R*8)  TEMAX      = MAXIMUM  TEMPERATURE
C          (I*4)  NUMTE      = NUMBER OF TEMPERATURE
C
C          (R*8)  FNEMIN     = MINIMIUM DENSITY OF TABLE
```

```
C          (R*8)  FNEMAX    = MAXIMUM  DENSITY
C          (I*4)  NUMNE     = NUMBER OF DENSITIES
C
C
C          (R*8)  FUELMAS   = MASS OF FUEL     (AMU)
C
C          (I*4)  IZ0       =  NUCLEAR CHARGE
C          (I*4)  IZL       =  LOWEST INCLUDED ION
C          (I*4)  IZU       =  HIGHEST INCLUDED ION
C
C          (I*4)  IZRA()    =  RECOMBINING ION (RAD.  RECOM.)
C          (I*4)  IZDA()    =  RECOMBINING ION (DIEL. RECOM.)
C          (I*4)  IZIA()    =  IONISING ION    (COLL. IONIS.)
C          (I*4)  IZTA()    =  RADIATING ION   (TOTAL LINE POWER)
C          (I*4)  IZSA()    =  RADIATING ION   (SPECIFIC LINE POWER)
C
C
C
C          (C*5)  CRRCA()   =  RADIATIVE RECOM. CODE
C          (I*4)  NRRCA()   =    - NOT USED -
C          (I*4)  ISRRCA()  =    - NOT USED -
C
C          (I*4)  NZA()     = LOWEST ACCESSIBLE SHELL FOR RAD. RECOM.
C          (I*4)  KSIA()    = NUMBER OF ELECTRONS IN SHELL
C
C          (I*4)  N0RA()    = LOWEST ACCESSIBLE PRINC. QUANTUM SHELL
C                             FOR RAD. RECOM.
C          (R*8)  V0RA()    = EFFECTIVE PRINCIPAL QUANTUM NUMBER
C                             FOR SHELL
C          (R*8)  PHFCRA()  = PHASE SPACE OCCUPANCY AVAILABILITY
C                             FOR SHELL
C          (R*8)  EDSPRA()  = ENERGY ADJUSTMENT IN LOWEST SHELL
C                             RATE COEFFICIENT
C          (R*8)  SCLERA()  = MULTIPLIER FOR LOWEST SHELL
C                             RATE COEFFICIENT
C
C
C
C          (C*5)  CDRCA()   =  DIELECTRONIC RECOM. CODE
C          (I*4)  NDRCA()   =  NUMBER OF TRANSITIONS FOLLOWING
C          (I*4)  ISDRCA()  =    - NOT USED -
C
C          (R*8)  DEDA(,)   = TRANSITION ENERGY (EV)
C          (R*8)  FDA(,)    = OSCILLATOR STRENGTH
C          (R*8)  GDA(,)    = GAUNT FACTOR
C          (I*4)  NNDA(,)   = DELTA N FOR TRANSITION
C          (I*4)  MSDA(,)   = MERTZ SWITCH (0=OFF, 1=ON)
C
C          (I*4)  ITYPDA(,) = TYPE OF DIELECTRONIC TRANSITION
C          (I*4)  N0DA(,)   = LOWEST ACCESSIBLE PRINC. QUANTUM SHELL
C                             FOR DIEL. RE
C          (I*4)  NCUTA(,)  = CUT-OFF PRINC. QUANTUM SHELL IN
C                             GENERAL PROGRAM
C          (I*4)  V0DA(,)   = EFFECTIVE PRINC. QUANTUM NUMBER
C                             FOR LOWEST ACCESS
C          (R*8)  PHFCDA(,) = PHASE SPACE OCCUPANCY AVAILABILITY
C                             FOR LOWEST SHELL
C          (R*8)  CRFCDA(,) = ADJUSTMENT FOR BETHE CORRECTIONS
C                             IN GENERAL PROGRAM
C          (R*8)  EPSIJA(,) = Z-SCALED PARENT TRANSITION ENERGY (RYD)
C          (R*8)  FIJA(,)   = OSCILLATOR STRENGTH FOR TRANSITION
```

```
C          (R*8)  EDSPDA(,) = ENERGY ADJUSTMENT IN BURGESS GENERAL
C                              FORMULA (RYD)
C          (R*8)  SCLEDA(,) = MULTIPLIER ON BURGESS GENERAL FORMULA
C
C
C
C          (C*5)  CCIOA()   =  COLLISIONAL IONIS. CODE
C          (I*4)  NCIOSA()  =  NUMBER OF SHELL  VALUES FOLLOWING
C          (I*4)  NCIORA()  =  NUMBER OF RESON. VALUES FOLLOWING
C          (I*4)  ISCIOA()  =    - NOT USED -
C
C          (R*8)  PIOA(,)   = SHELL IONISATION POTENTIAL (EV)
C          (R*8)  AIOA(,)   = LOTZ PARAMETER
C          (R*8)  BIOA(,)   = LOTZ PARAMETER
C          (R*8)  CIOA(,)   = LOTZ PARAMETER
C          (I*4)  NQIOA(,)  = EQUIVALENT ELECTRONS IN SHELL
C
C          (R*8)  ZETAA(,)  = NUMBER OF EQUIVALENT ELECTRONS FOR SHELL
C          (R*8)  EIONA(,)  = IONISATION ENERGY FOR SHELL (RYD)
C          (R*8)  CIA(,)    = MULTIPLIER FOR BURGESS-CHIDICHIMO RATE
C                              FOR SHELL
C          (R*8)  WGHTA(,)  = WEIGHTING FACTOR FOR EXCITATION TO
C                              RESONANCE
C          (R*8)  ENERA(,)  = EXCITATION ENERGY FOR TRANSITION
C                              TO RESONANCE (RYD)
C          (R*8)  CRA(,)    = MULTIPLIER ON EXCITATION RATE EXPRESSSION
C
C
C
C          (C*5)  CPLTA()   =  TOTAL LINE POWER CODE
C          (I*4)  NPLTA()   =  NUMBER OF TRANSITIONS FOLLOWING
C          (I*4)  ISPLTA()  =    - NOT USED -
C
C          (R*8)  DEPTA(,)  = TRANSITION ENERGY (EV)
C          (R*8)  FPTA(,)   = OSCILLATOR STRENGTH
C          (R*8)  GPTA(,)   = GAUNT FACTOR
C          (I*4)  NNPTA(,)  = DELTA N FOR TRANSITION
C
C          (R*8)  SPYLTA(,) = MULTIPLIER OF VAN REGEMORTER P
C                              FACTOR IN TOTAL POWER
C
C
C
C          (C*5)  CPLSA()   = SPECIFIC LINE POWER CODE
C          (I*4)  NPLSA()   =   - NOT USED -
C          (I*4)  ISPLSA()  =   - NOT USED -
C          (C*8)  INFO()    = WAVELENGTH OF SPECIFIC LINE FOR
C                              NAMING PURPOSES
C
C          (R*8)  DEPSA(,)  = TRANSITION ENERGY (EV)
C          (R*8)  FPSA(,)   = OSCILLATOR STRENGTH
C          (R*8)  GPSA(,)   = GAUNT FACTOR
C          (I*4)  NNPSA(,)  = DELTA N FOR TRANSITION
C
C          (R*8)  SPYLSA(,) = MULTIPLIER OF VAN REGEMORTER P FACTOR
C                              IN SPECIFIC LINE POWER
C
C
C
C  ROUTINES:
C          ROUTINE   SOURCE     DESCRIPTION
```

```
C          ------------------------------------------------------
C          I4UNIT    ADAS       FETCH UNIT NUMBER FOR MESSAGE OUTPUT
C          D8FLIN    ADAS408    INITIALISE ENERGY MESH FOR FILTER
C                               INTEGRATION
C          D8TRAN    ADAS408    FILTER TRANSMISSION AT PARTICULAR ENERGY
C          D8INTG    ADAS408    INTEGRATE WITH FILTER
C          D8CXSC    ADAS408    RETURNS CROSS-SECTIONS FROM CX COLLISION
C          D8VGOL    ADAS408    CALCULATES VON GOELER RECOMB. COEFFS
C          NGFFMH    ADAS       FREE-FREE GAUNT FACTOR
C
C
C
C  HISTORY:  DERIVED FROM NCRAT0  --- J. SPENSE,     TESSELLA
C                                     H. P. SUMMERS, JET
C                                     27/3/1990
C
C  CHANGES : 13/12/90  H.P.SUMMERS - CHANGE CONSTANTS FOR LINE POWER AND
C                                    BREMSSTRAHLUNG POWER TO MATCH
C                                    ABELS-VAN MAANEN (1985).
C                                    NB. NO MERTZ SWITCH ON H-LIKE
C                                    AND HE-LIKE DIELECTRONIC RECOMB.
C          :  5/ 2/91  H.P.SUMMERS - ALTER COMMON /RATCOM/ TO REMOVE
C                                    UNNECESSARY STORAGE AND ORGANISE
C                                    FOR NEW (ADF03) DATA INPUT.
C          :  1/ 8/91  H.P.SUMMERS - ALTER LINFO DIMENSION TO ALLOW
C                                    IT AS AN INFORMATION STRING FOR ALL
C                                    DATA SETS.  SET THE HYDROGEN
C                                    ISOTOPE MASS IN LINFO AS
C                                    ' MH=*.**' FOR CCD AND PRC.
C                                    ADD HMADAS TO /RATCOM/
C          :  6/8/92   M O'MULLANE - STAND-ALONE VERSION
C
C
C
C  ADAS408 IMPLEMENTATION
C
C  AUTHOR: M O'MULLANE, UCC
C
C  DATE:    10/05/94
C
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 15-04-96
C MODIFIED: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST CONVERTED
C
C VERSION: 1.2 DATE: 14-05-96
C MODIFIED: WILLIAM OSBORN
C     REARRANGED ARGUMENTS TO STAY UNDER
C     LIMIT OF 20 CONTINUATION CHARACTERS AT ARCETRI AND GARCHING
C
C VERSION: 1.3 DATE: 14-05-96
C MODIFIED: WILLIAM OSBORN
C           S.C.C.S. ERROR
C
C VERSION: 1.4 DATE: 30-01-98
C MODIFIED: Martin O'Mullane
C           Added free-free Gaunt factor to the eveluation of bremsstrahlung
C           power. Used NGFFMH to calculate gff.
C
```

```
C VERSION: 1.5 DATE: 05-08-2003
C MODIFIED: Martin O'Mullane
C            - ADAS408 restructured.
C            - Pass in adf03 and adf35 data files rather than the
C              data in them.
C
C VERSION : 1.6
C DATE    : 05-03-2004
C MODIFIED: Martin O'Mullane
C             - Alter dimensions for 55 temperatures, 50 densities,
C               80 ion stages, and 15 groups.
C             - Type B ionisation was wrong. Calculate it with a call
C               to rbchid.for.
C             - Warn when EA contribution to type B ionisation is
C               attempted as it is not coded.
C             - Warn when type B radiative recombination is
C               requested as the prb is not yet calculated.
C             - Pass out separated RR and DR recombination rates.
C             - Pass out separated RR, DR and BR power coefficients
C               but only for the non-filtered case.
C
C VERSION : 1.7
C DATE    : 06-10-2004
C MODIFIED: Martin O'Mullane
C             - Increase number of groups in the adf03 file to 20
C               from 15 to cope with high Z DR.
C
C VERSION : 1.8
C DATE    : 24-06-2005
C MODIFIED: Martin O'Mullane
C            - Add arrays for filtered rr/dr/br power.
C
C-----------------------------------------------------------------------
      CHARACTER*80        DSN03,       DSN35
      CHARACTER*8         INFCAL(IZDIMD),           INFPLS(IZDIMD)
      CHARACTER*8         INFPLT(IZDIMD),           INFPLTN(IZDIMD)
      CHARACTER*8         INFPRB(IZDIMD),           INFPRBN(IZDIMD)
      CHARACTER*8         INFPRC(IZDIMD),           INFPRCN(IZDIMD)
      CHARACTER*8         INFRAL(IZDIMD),           INFSAL(IZDIMD)
      CHARACTER*24        METCAL,      METPLS,      METPLT,      METPLTN
      CHARACTER*24        METPRB,      METPRBN,     METPRC,      METPRCN
      CHARACTER*24        METRAL,      METSAL
      CHARACTER*13        NAME
      INTEGER             IZ0,         IZL,         IZU,         NUMNE
      INTEGER             NUMTE
      LOGICAL             LTICK
      REAL*8              CAL(ITDIMD,IDDIMD,IZDIMD)
      REAL*8              FNEL(ITDIMD),             FNEMAX,      FNEMIN
      REAL*8              FUELMAS,     PLSL(ITDIMD,IDDIMD,IZDIMD)
      REAL*8              PLSL0(ITDIMD,IDDIMD)
      REAL*8              PLTL(ITDIMD,IDDIMD,IZDIMD)
      REAL*8              PLTL0(ITDIMD,IDDIMD)
      REAL*8              PLTL0NFL(ITDIMD,IDDIMD)
      REAL*8              PLTLNFL(ITDIMD,IDDIMD,IZDIMD)
      REAL*8              PRBBR(ITDIMD,IDDIMD,IZDIMD)
      REAL*8              PRBBRNFL(ITDIMD,IDDIMD,IZDIMD)
      REAL*8              PRBDR(ITDIMD,IDDIMD,IZDIMD)
      REAL*8              PRBDRNFL(ITDIMD,IDDIMD,IZDIMD)
      REAL*8              PRBL(ITDIMD,IDDIMD,IZDIMD)
      REAL*8              PRBLNFL(ITDIMD,IDDIMD,IZDIMD)
      REAL*8              PRBRR(ITDIMD,IDDIMD,IZDIMD)
```

```
REAL*8             PRBRRNFL(ITDIMD,IDDIMD,IZDIMD)
REAL*8             PRCL(ITDIMD,IDDIMD,IZDIMD)
REAL*8             PRCLNFL(ITDIMD,IDDIMD,IZDIMD)
REAL*8             RAL(ITDIMD,IDDIMD,IZDIMD)
REAL*8             RALDR(ITDIMD,IDDIMD,IZDIMD)
REAL*8             RALRR(ITDIMD,IDDIMD,IZDIMD)
REAL*8             SAL(ITDIMD,IDDIMD,IZDIMD)
REAL*8             SAL0(ITDIMD,IDDIMD),      TEL(ITDIMD), TEMAX
REAL*8             TEMIN
```

5.53 d8gpca: Subroutine d8gpca from library adas4xx

```
       SUBROUTINE D8GPCA  (TEA    , IZ1    ,
     &                     ITYPE  , N0     , V0     ,
     &                     EIJ_in , FIJ    , EDISPG , SCALEG ,
     &                     PHFRAC , CORFAC , NCUT_in, NG ,
     &                     ALF )

C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: D8GPCA *********************
C
C PURPOSE :  ROUTINE TO PROVIDE BURGESS GENERAL PROGRAM RESULTS
C            AT A GIVEN TEMPERATURES AND AT ZERO DENSITY.
C
C            EQUAL THE GENERAL FORMULA RESULTS AS FAR AS POSSIBLE
C            BY MODIFICATION OF BETHE CORRECTIONS VIA A SINGLE
C            SCALING PARAMETER CORFAC.
C            THE CORRECTION FACTORS USED IN THE GENERAL PROGRAM
C            ARE OBTAINED BY ADJUSTMENT OF STANDARD SETS FOR SPECIFIC
C            TYPES OF TRANSITION.  THE ADJUSTMENT IS
C            (NEW COR(J))=EXP(-CORFAC/(L**DF+0.5))*(STANDARD COR(J)
C            THE STANDARD COR'S ARE AS FOLLOWS:
C
C  TYPE    TRANSITION                    COR'S                      DF
C    1   NI=1,NJ>=2,LJ=LI+1:       0.05,0.30,0.50,0.90             2.0
C    2   NI=2,NJ=3,LJ=LI+1:        0.01,0.02,0.20,0.40,0.70,0.90   1.0
C    3   NI=2,NJ=3,LJ=LI-1:        0.01,0.01,0.01,0.08,0.30,0.70   1.0
C    4   NJ-NI=0, LJ=LI+1 :        0.30,0.35,0.40,0.45,0.70,0.90   0.5
C    5   NJ-NI=0, LJ=LI-1 :        0.30,0.35,0.40,0.45,0.70,0.90   0.5
C    6   NJ-NI>0, LJ=LI+1 :        0.01,0.02,0.20,0.40,0.70,0.90   1.0
C    7   NJ-NI>0, LJ=LI-1 :        0.01,0.01,0.01,0.08,0.30,0.70   1.0
C
C  CALLING PROGRAM: ADAS408
C
C
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  IZ1       = RECOMBINING ION CHARGE
C
C          (I*4)  ITYPE     = TYPE OF DIELECTRONIC TRANSITION
C          (I*4)  N0        = LOWEST ACCESSIBLE PRINC. QUANTUM SHELL
C                             FOR DIEL. RE
C          (I*4)  NCUT      = CUT-OFF PRINC. QUANTUM SHELL IN
C                             GENERAL PROGRAM
C          (I*4)  NG        = CUT-OFF PRINC. QUANTUM SHELL FROM
C                             COLLISINAL IONISATION
C          (R*8)  V0        = EFFECTIVE PRINC. QUANTUM NUMBER
C                             FOR LOWEST ACCESS
C          (R*8)  PHFRAC    = PHASE SPACE OCCUPANCY AVAILABILITY
C                             FOR LOWEST SHELL
C          (R*8)  CFAC      = ADJUSTMENT FOR BETHE CORRECTIONS
C                             IN GENERAL PROGRAM
C          (R*8)  EIJ       = Z SCALED PARENT TRANSITION ENERGY (RYD)
C          (R*8)  FIJ       = OSCILLATOR STRENGTH FOR TRANSITION
C          (R*8)  EDSP      = ENERGY ADJUSTMENT IN BURGESS GENERAL
C                             FORMULA (RYD)
C          (R*8)  SCALE     = MULTIPLIER ON BURGESS GENERAL FORMULA
C
C          (R*8)  TEA       = TEMPERATURE OF CALCULATION (K)
```

```
C
C
C  OUTPUT: (R*8)  ALFO      = GENERAL PROGRAM DIELECTRONIC COEFFICIENTS
C          (R*8)  PHFRAC    = REVISED PHASE SPACE FACTOR
C          (R*8)  CORFAC    = REVISED BETHE CORRECTION SCALER
C
C
C  PROGRAM:
C
C
C  ROUTINES:
C          ROUTINE   SOURCE   DESCRIPTION
C          ------------------------------------------------------
C          GPDIEL    ADAS     ?
C          BF        ADAS     ?
C
C
C  HISTORY : BASED ON GPCALC
C            H P SUMMERS   11-5-87
C
C  AUTHOR: M O'MULLANE, UCC
C
C  DATE:    28/07/94
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 15-04-96
C MODIFIED: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST CONVERTED
C
C VERSION: 1.2                          DATE: 23-05-96
C MODIFIED: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C               REPLACED CFAC WITH CORFAC: ERROR. REMOVED ALF0 AND ALFDAT
C
C
C VERSION: 1.3                          DATE: 16-01-2004
C MODIFIED: Martin O'Mullane
C             - Added ncut_in and eij_in as input arguments because
C               ncut and eij are altered in this subroutine.
C             - Trap for AD1.eq.0 for shells above n=10.
C             - X was defined the same way for the for both parts.
C               Re-diefine it for n>10 using v1 rather than v.
C
C VERSION: 1.4                          DATE: 17-05-2007
C MODIFIED: Allan Whiteford
C             - Updated comments as part of subroutine documentation
C               procedure.
C
C-----------------------------------------------------------------------
      INTEGER             ITYPE,       IZ1,         N0,          NCUT_IN
      INTEGER             NG
      REAL*8              ALF,         CORFAC,      EDISPG,      EIJ_IN
      REAL*8              FIJ,         PHFRAC,      SCALEG,      TEA
      REAL*8              V0
```

5.54 d8intg: Subroutine d8intg from library adas4xx

```
       subroutine d8intg( ndedge  ,  ndeng   ,
     &                    iedge   ,  ieng    ,
     &                    edge    ,  energy  ,  fraction ,
     &                    te      ,  flimit  ,  result
     &                  )

C-----------------------------------------------------------------------
C
C  ******************** FORTRAN77 SUBROUTINE: D8TRAN ******************
C
C  PURPOSE: To integrate between a and b with an interval of step
C           the integrand
C                           f(x)exp(-x) * exp(+a)
C           where f(x) is the filter function.
C
C
C  CALLING PROGRAM: adas408
C
C  FUNCTION:
C
C  input : (i*4)  ndeng    = maximum number of energies in adf35 file.
C  input : (i*4)  ndedge   = maximum number of energy edges in adf35 file.
C  input : (i*4)  ieng     = actual number of energies.
C  input : (i*4)  iedge    = actual number of edges.
C  input : (r*8)  edge     = tabulated edge energies (eV).
C  input : (r*8)  energy   = tabulated energies (eV).
C  input : (r*8)  fraction = tabulated transmission fractions.
C  input : (r*8)  te       = user supplied temperature (eV).
C  input : (r*8)  flimit   = lower limit of integration (eV/Te)
C
C  output: (r*8)  result   = value of integral.
C
C  NOTES:
C
C  ROUTINES:
C           ROUTINE    SOURCE    BRIEF DESCRIPTION
C           ------------------------------------------------------------
C           i4indfvs   ADAS      Finds nearest index for a non-monotonic
C                                array
C           xxmerg     ADAS      Merge two grids.
C           d8tran     ADAS      Returns transmission of a filter.
C           d8part     ADAS      Trapezoidal integration routine.
C
C  Author   : Martin O'Mullane UCC 26/8/92
C
C  VERSION  : 1.1
C  DATE     : 15-04-96
C  MODIFIED : Martin O'Mullane
C               - First version in SCCS.
C
C  VERSION  : 1.2
C  DATE     : 05-0-2003
C  MODIFIED : Martin O'Mullane
C               - Uses adf35 filter file data.
C
C  VERSION  : 1.3
C  DATE     : 16-02-2005
C  MODIFIED : Martin O'Mullane
C               - Do not re-use x1() and x2() in parts integration.
```

```
C
C
C-----------------------------------------------------------------------
      INTEGER             IEDGE,       IENG,        NDEDGE,      NDENG
      REAL*8              EDGE(NDEDGE),             ENERGY(NDENG)
      REAL*8              FLIMIT,      FRACTION(NDENG),          RESULT
      REAL*8              TE
```

5.55 d8part: Subroutine d8part from library adas4xx

```
       subroutine d8part(x, y, num, te, flimit, result)

C-----------------------------------------------------------------------
C
C  ******************** FORTRAN77 SUBROUTINE: D8TRAN ******************
C
C  PURPOSE: To integrate between x(1) and x(num) the integrand
C                           f(x)exp(-x) * exp(+a)
C           where f(x) is the filter function.
C
C
C  CALLING PROGRAM: adas408
C
C  FUNCTION:
C
C  input : (r*8)  x        = tabulated edge energies (eV).
C  input : (r*8)  y        = tabulated energies (eV).
C  input : (i*4)  num      = actual number of edges.
C  input : (r*8)  te       = user supplied temperature (eV).
C  input : (r*8)  flimit   = lower limit of integration (eV/Te)
C
C  output: (r*8)  result   = value of integral.
C
C  NOTES:
C
C  ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C
C
C  VERSION  : 1.1
C  DATE     : 05-08-2003
C  MODIFIED : Martin O'Mullane
C              - First version in SCCS.
C
C  VERSION  : 1.2
C  DATE     : 16-02-2005
C  MODIFIED : Martin O'Mullane
C              - Do not re-use x1() and x2() in parts integration.
C
C-----------------------------------------------------------------------

C-----------------------------------------------------------------------
      INTEGER             NUM
      REAL*8              FLIMIT,      RESULT,      TE,          X(*)
      REAL*8              Y(*)
```

5.56 d8tran: Subroutine d8tran from library adas4xx

```
       subroutine d8tran(ndeng  , ndedge  ,
     &                   ieng   , iedge   ,
     &                   edge   , energy  , fraction ,
     &                   ein    , fout
     &                  )

C-----------------------------------------------------------------------
C
C  ******************** FORTRAN77 SUBROUTINE: D8TRAN ******************
C
C  PURPOSE:  To determine transmission fraction at energy ein.
C
C  CALLING PROGRAM: adas408
C
C  FUNCTION:
C
C  input : (i*4)  ndeng    = maximum number of energies in adf35 file.
C  input : (i*4)  ndedge   = maximum number of energy edges in adf35 file.
C  input : (i*4)  ieng     = actual number of energies.
C  input : (i*4)  iedge    = actual number of edges.
C  input : (r*8)  edge     = tabulated edge energies (eV).
C  input : (r*8)  energy   = tabulated energies (eV).
C  input : (r*8)  fraction = tabulated transmission fractions.
C  input : (r*8)  ein      = user supplied energy (eV).
C
C  output: (r*8)  fout     = transmission fraction at ein.
C
C  NOTES: No extrapolation is allowed and energies outside the range
C         and set to the limit values.
C
C  ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          i4indfvs   ADAS      Finds nearest index for a non-monotonic
C                               array
C          xxpint     ADAS      Order 3 polynomial interpolation.
C
C
C  VERSION  : 1.1
C  DATE     : 15-04-96
C  MODIFIED : Martin O'Mullane
C              - First version.
C
C  VERSION  : 1.2
C  DATE     : 23-07-2003
C  MODIFIED : Martin O'Mullane
C              - Interpolates adf35 filter file data
C                rather than calculating the fraction from formulae.
C                This allows a wider range of filters.
C
C  VERSION  : 1.3
C  DATE     : 16-02-2005
C  MODIFIED : Martin O'Mullane
C              - Drop warnings to screen.
C
C-----------------------------------------------------------------------
      INTEGER             IEDGE,       IENG,        NDEDGE,      NDENG
      REAL*8              EDGE(NDEDGE),             EIN
      REAL*8              ENERGY(NDENG),            FOUT
```

```
REAL*8              FRACTION(NDENG)
```

5.57 d8vgol: Subroutine d8vgol from library adas4xx

```
      subroutine d8vgol(tea, ga, ga0, garest, z1, n0, v0, phfrac)


C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: D8VGOL *********************
C
C  PURPOSE:    Routine to evaluate total radiative recombination rate
C              coefficients at zero density using the Von Goeler type
C              formula with modified capture to the lowest accessible
C              principal quantum shell.
C
C              PHFRAC gives the proportion of the lowest level capture
C              allowed based on the available phase space of occupied
C              shells arguments.
C
C  CALLING PROGRAM: ADAS408
C
C
C  INPUT:  (R*8)  TEA         = Electron temperatures (k)
C          (R*8)  z1          = Recombining ion charge
C          (I*4)  n0          = Lowest accessible n-shell by recombination
C          (R*8)  v0          = Effective principal quantum number of
C                               lowest accessible shell
C          (R*8)  phfrac      = Phase space occupation factor for lowest
C                               accessible shell
C
C
C  OUTPUT: (R*8)  ga          = Total radiative recombination
C                               coefficient (cm**3 sec-1)
C          (R*8)  ga0         = Ground shell  recombination coefficient
C          (R*8)  garest      = Recombination coefficient to all shells
C                               excluding the ground shell.
C
C
C  ROUTINES:
C          ROUTINE   SOURCE     DESCRIPTION
C          ---------------------------------------------------------
C
C
C  HISTORY:
C
C  H.P. Summers, JET       24 June 1987
C  M. O'Mullane            10 Aug  1992 - modified for one temperature
C
C
C-----------------------------------------------------------------------
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 15-04-96
C MODIFIED: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST CONVERTED
C
C VERSION: 1.2 DATE: 16-02-2004
C MODIFIED: Martin O'Mullane
C            - Convert to implicit none.
C
C VERSION: 1.3 DATE: 17-05-2007
C MODIFIED: Allan Whiteford
```

```
C            - Updated comments as part of subroutine documentation
C              procedure.
C
C-----------------------------------------------------------------------
      INTEGER             N0
      REAL*8              GA,          GA0,         GAREST,      PHFRAC
      REAL*8              TEA,         V0,          Z1
```

5.58 d8wzcd: Subroutine d8wzcd from library adas4xx

```
      subroutine d8wzcd( iuntz     , iunty     , iunte   ,
     &                   lzcd      , lycd      , lecd    ,
     &                   itdimd    , iddimd    ,
     &                   izdimd    , iodimd    , imdimd  ,
     &                   itmax     , idmax     ,
     &                   dtev      , ddens     ,
     &                   iz0       , izl       , izu     ,
     &                   user      , date
     &                 )

C-----------------------------------------------------------------------
C
C  ****************** fortran77 subroutine: d8wzcd *********************
C
C  purpose:  To create zcd, ycd and ecd files for unresolved baseline.
C
C  calling program: adas408
C
C
C  subroutine:
C
C  input : (i*4)  iuntz     = unit for zcd file output
C  input : (i*4)  iunty     = unit for ycd file output
C  input : (i*4)  iunte     = unit for ecd file output
C  input : (i*4)  iunte     = unit for ecd file output
C  input : (l*4)  lzcd      = .true.  => output zcd file
C                             .false. => do not output zcd file
C  input : (l*4)  lycd      = .true.  => output ycd file
C                             .false. => do not output ycd file
C  input : (l*4)  lecd      = .true.  => output ecd file
C                             .false. => do not output ecd file
C  input : (i*4)  itdimd    = maximum number of temperatures
C  input : (i*4)  iddimd    = maximum number of densities
C  input : (i*4)  izdimd    = maximum number of charge states
C  input : (i*4)  itmax     = number of temperatures
C  input : (i*4)  idmax     = number of densities
C  input : (r*8)  dtev()    = temperature set of tables (ev) - log mesh
C  input : (r*8)  ddens()   = density set of tables (cm-3) - log mesh
C  input : (i*4)  iz0       = nuclear charge
C  input : (i*4)  izl       = first included ion (=0 for neutral)
C  input : (i*4)  izu       = last included ion (=iz0 for bare nucleus)
C  input : (c*30) user      = producer
C  input : (c*8)  date      = date string.
C
C  routines:
C          routine    source    brief description
C          ------------------------------------------------------------
C          i4unit     adas      fetch unit number for output of messages
C          xxword     adas      parses a string into separate words
C          xxopen     adas      check existence and open a file
C          xxrmve     adas      removes occurrences of a char. in string
C          xxmkrp     adas      create the root partition text lines
C          xxslen     adas      finds the length of a string excluding
C                               leading and trailing blanks
C          xfesym     adas      fetch the chemical symbol of an element
C          xfelem     adas      fetch the name of an element
C          xxdata_00  adas      read an adf00 dataset
C
C
```

```
C author:  H. P. Summers, university of strathclyde
C          ja7.08
C          tel. 0141-548-4196
C
C date:    06/10/06
C
C
C version  : 1.1
C date     : 06-10-2006
C modified : Hugh Summers
C              - first version
C
C version  : 1.2
C date     : 16-01-2007
C modified : Hugh Summers
C              - adjustment to ecd part to include z1=0 quasi-state
C                for the neutral creation energy. Use new version of
C                xxdata_00.for to handle metastable resolved cases.
C
C version  : 1.3
C date     : 08-03-2007
C modified : Hugh Summers
C              - adjustment of first output file line to include adf no
C                and remove class from ion header lines.
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      CHARACTER*8         DATE
      CHARACTER*30        USER
      INTEGER             IDDIMD,      IDMAX,       IMDIMD,      IODIMD
      INTEGER             ITDIMD,      ITMAX,       IUNTE,       IUNTY
      INTEGER             IUNTZ,       IZ0,         IZDIMD,      IZL
      INTEGER             IZU
      LOGICAL             LECD,        LYCD,        LZCD
      REAL*8              DDENS(IDDIMD),            DTEV(ITDIMD)
```

5.59 d9data: Subroutine d9data from library adas4xx

```
C Copyright (c) 1997, Strathclyde University.
       SUBROUTINE D9DATA( DSFLLA , LSELA  , LEXSA  , LDEFA , LPART  ,
     &                    IZ0    , IZ1MIN , IZ1MAX , NPART  ,
     &                    NTDIM  , NDDIM  , ITMAX  , IDMAX  ,
     &                    ISDIMD , IZDIMD , ITDIMD , IPDIMD , NPARTR,
     &                    DTEV   , DDENS  ,
     &                    DTEVD  , DDENSD , DRCOFD , ZDATA  ,
     &                    DRCOFI ,
     &                    ACDA   , LACDA  ,
     &                    SCDA   , LSCDA  ,
     &                    CCDA   , LCCDA  ,
     &                    PRBA   , LPRBA  ,
     &                    PRCA   , LPRCA  ,
     &                    QCDA   , LQCDA  ,
     &                    XCDA   , LXCDA  ,
     &                    PLTA   , LPLTA
     &                  )
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: D9DATA *********************
C
C PURPOSE : TO EXTRACT A COMPLETE SET OF COLLISIONAL DIELECTRONIC DATA
C           FOR A (TEMPERATURE, DENSITY) GRID
C           FROM EITHER PARTIAL (METASTABLE/PARENT RESOLVED) OR STANDARD
C           (UNRESOLVED) ISONUCLEAR MASTER FILES
C
C           DERIVED FROM D5DATA
C
C NOTE    : THE SOURCE DATA IS CONTAINED AS SEQUENTIAL DATASETS
C           WITH THE FOLLOWING NAMING CONVENTIONS:
C
C                   (1) JETSHP.ACD<YR>#<EL).<CODE>DATA
C                   (2) JETSHP.SCD<YR>#<EL>.<CODE>DATA
C                   (3) JETSHP.CCD<YR>#<EL>.<CODE>DATA
C                   (4) JETSHP.PRB<YR>#<EL>.<FILT>.<CODE>DATA
C                   (5) JETSHP.PRC<YR>#<EL>.<FILT>.<CODE>DATA
C                   (6) JETSHP.QCD<YR>#<EL>.<CODE>DATA
C                   (7) JETSHP.XCD<YR>#<EL>.<CODE>DATA
C                   (8) JETSHP.PLT<YR>#<EL>.<CODE>DATA
C
C       WHERE, <YR>   = TWO DIGIT YEAR NUMBER
C              <EL>   = ONE OR TWO CHARACTER ELEMENT SYMBOL
C              <CODE> = R       => PARTIAL DATA
C                       U       => PARTIAL DATA
C                       OMITTED => STANDARD DATA
C              <FILT> = SIX CHARACTER POWER FILTER CODE
C
C       AND DATA OF CLASSES 6 AND 7 DO NOT EXIST FOR THE STANDARD CASE.
C
C
C INPUT  : (C*120)DSFLLA()  = MASTER FILE DATA SET NAMES (FULL MVS DSN)
C                             (IN FORM SUITABLE FOR DYNAMIC ALLOCATION)
C INPUT  : (L*4)  LSELA()   = .TRUE.  => INPUT DATA SET TYPE FOR THIS
C                                        INDEX SELECTED
C                           = .FALSE. => INPUT DATA SET FOR THIS INDEX
C                                        NOT SELECTED
C INPUT  : (L*4)  LEXSA()   = .TRUE.  => INPUT DATA SET TYPE FOR THIS
C                                        SELECTED INDEX EXISTS
```

```
C                              = .FALSE. => INPUT DATA SET DOES NOT EXIST
C                                           FOR THIS SELECTED INDEX
C INPUT  : (L*4)  LDEFA()      = .TRUE.  => INPUT DATA SET TYPE FOR THIS
C                                           DEFAULT YEAR INDEX EXISTS
C                              = .FALSE. => INPUT DATA SET DOES NOT EXIST
C                                           FOR THIS DEFAULT YEAR INDEX
C INPUT  : (L*4)  LPART        = .TRUE.  => PARTIAL DATA SELECTED
C                              = .FALSE. => STANDARD DATA SELECTED
C INPUT  : (I*4)  IZ0          = NUCLEAR CHARGE
C INPUT  : (I*4)  IZ1MIN       = MINIMUM ION CHARGE+1 IN MASTER DATA FILES
C INPUT  : (I*4)  IZ1MAX       = MAXIMUM ION CHARGE+1 IN MASTER DATA FILES
C INPUT  : (I*4)  NPART()      = METASTABLE PARTITION.  I.E. NUMBER OF
C                                METASTABLES FROM CHARGE STATE IZ1MIN-1 TO
C                                IZ1MAX ON INPUT
C INPUT  : (I*4)  NTDIM        = MAXIMUM NUMBER OF DTEV VALUES
C INPUT  : (I*4)  NDDIM        = MAXIMUM NUMBER OF DDENS VALUES
C INPUT  : (I*4)  ITMAX        = NUMBER OF DTEV() VALUES
C INPUT  : (I*4)  IDMAX        = NUMBER OF DDENS() VALUES
C INPUT  : (I*4)  ISDIMD       = MAXIMUM NUMBER OF (CHARGE, PARENT, GROUND)
C                                BLOCKS IN ISONUCLEAR MASTER FILES
C INPUT  : (I*4)  IZDIMD       = MAXIMUM NUMBER OF CHARGE STATES
C                                IN ISONUCLEAR MASTER FILES
C INPUT  : (I*4)  ITDIMD       = MAXIMUM NUMBER OF TEMP OR DENS VALUES IN
C                                ISOELECTRONIC MASTER FILES
C INPUT  : (I*4)  IPDIMD       = MAXIMUM NUMBER OF METASTABLES FOR EACH
C                                IONISATION STAGE
C INPUT  : (R*8)  DTEV()       = DLOG10(ELECTRON TEMPERATURES (EV))
C INPUT  : (R*8)  DDENS()      = DLOG10(ELECTRON DENSITIES (CM-3))
C
C OUTPUT : (I*4)  NPARTR()     = METASTABLE PARTITION.  I.E. NUMBER OF
C                                METASTABLES FROM CHARGE STATE IZ1MIN-1 TO
C                                IZ1MAX FOUND IN MASTER FILE
C OUTPUT : (R*8)  DTEVD()      = DLOG10(DATA ELECTRON TEMPERATURES (EV))
C                                IN SELECTED MASTER FILE
C OUTPUT : (R*8)  DDENSD()     = DLOG10(DATA ELECTRON DENSITIES (CM-3))
C                                IN SELECTED MASTER FILE
C OUTPUT : (R*8)  DRCOFD(,,)= DLOG10(DATA RATE COEFFICIENTS (CM-3/S))
C                                IN SELECTED MASTER FILE
C                                1ST DIM: (CHARGE,META,GRD) BLOCK INDEX
C                                2ND DIM: TEMPERATURE INDEX
C                                3RD DIM: DENSITY INDEX
C OUTPUT : (R*8)  ZDATA()      = CHARGE + 1 FOR IONS IN SELECTED MASTER
C                                FILE
C                                1ST DIM: (CHARGE,META,GRD) BLOCK INDEX
C OUTPUT : (R*8)  DRCOFI(,) = INTERPOLATION OF DRCOFD(,,) FOR
C                                DTEV() & DDENS()
C OUTPUT : (R*8)  ACDA(,,,,)= INTERPOLATION OF ACD COEFFICIENT (CM3 S-1)
C                                1ST DIM: TEMPERATURE INDEX
C                                2ND DIM: DENSITY INDEX
C                                3RD DIM: CHARGE STATE INDEX
C                                4RD DIM: RECOMBINING METASTABLE INDEX
C                                5TH DIM: RECOMBINED METASTABLE INDEX
C OUTPUT : (L*4)  LACDA(,,) = .TRUE.  => ACD COEFFICIENT AVAILABLE
C                                .FALSE. => ACD COEFFICIENT NOT AVAILABLE
C                                1ST DIM: CHARGE STATE INDEX
C                                2ND DIM: RECOMBINING METASTABLE INDEX
C                                3RD DIM: RECOMBINED METASTABLE INDEX
C OUTPUT : (R*8)  SCDA(,,,,)= INTERPOLATION OF SCD COEFFICIENT (CM3 S-1)
C                                1ST DIM: TEMPERATURE INDEX
C                                2ND DIM: DENSITY INDEX
C                                3RD DIM: CHARGE STATE INDEX
```

```
C                               4RD DIM: RECOMBINING METASTABLE INDEX
C                               5TH DIM: RECOMBINED METASTABLE INDEX
C OUTPUT : (L*4)  LSCDA(,,) = .TRUE.  => SCD COEFFICIENT AVAILABLE
C                               .FALSE. => SCD COEFFICIENT NOT AVAILABLE
C                               1ST DIM: CHARGE STATE INDEX
C                               2ND DIM: RECOMBINING METASTABLE INDEX
C                               3RD DIM: RECOMBINED METASTABLE INDEX
C OUTPUT : (R*8)  CCDA(,,,,)= INTERPOLATION OF CCD COEFFICIENT (CM3 S-1)
C                               1ST DIM: TEMPERATURE INDEX
C                               2ND DIM: DENSITY INDEX
C                               3RD DIM: CHARGE STATE INDEX
C                               4RD DIM: RECOMBINING METASTABLE INDEX
C                               5TH DIM: RECOMBINED METASTABLE INDEX
C OUTPUT : (L*4)  LCCDA(,,) = .TRUE.  => CCD COEFFICIENT AVAILABLE
C                               .FALSE. => CCD COEFFICIENT NOT AVAILABLE
C                               1ST DIM: CHARGE STATE INDEX
C                               2ND DIM: RECOMBINING METASTABLE INDEX
C                               3RD DIM: RECOMBINED METASTABLE INDEX
C OUTPUT : (R*8)  PRBA(,,,) = INTERPOLATION OF PRB COEFFICIENT (W CM3 )
C                               1ST DIM: TEMPERATURE INDEX
C                               2ND DIM: DENSITY INDEX
C                               3RD DIM: CHARGE STATE INDEX
C                               4TH DIM: RECOMBINING METASTABLE INDEX
C OUTPUT : (L*4)  LCCDA(,)  = .TRUE.  => PRB COEFFICIENT AVAILABLE
C                               .FALSE. => PRB COEFFICIENT NOT AVAILABLE
C                               1ST DIM: CHARGE STATE INDEX
C                               2ND DIM: RECOMBINING METASTABLE INDEX
C OUTPUT : (R*8)  PRCA(,,,) = INTERPOLATION OF PRC COEFFICIENT (W CM3 )
C                               1ST DIM: TEMPERATURE INDEX
C                               2ND DIM: DENSITY INDEX
C                               3RD DIM: CHARGE STATE INDEX
C                               4TH DIM: RECOMBINING METASTABLE INDEX
C OUTPUT : (L*4)  LPRCA(,)  = .TRUE.  => PRC COEFFICIENT AVAILABLE
C                               .FALSE. => PRC COEFFICIENT NOT AVAILABLE
C                               1ST DIM: CHARGE STATE INDEX
C                               2ND DIM: RECOMBINING METASTABLE INDEX
C OUTPUT : (R*8)  QCDA(,,,,)= INTERPOLATION OF QCD COEFFICIENT (CM3 S-1)
C                               1ST DIM: TEMPERATURE INDEX
C                               2ND DIM: DENSITY INDEX
C                               3RD DIM: CHARGE STATE INDEX
C                               4TH DIM: FIRST METASTABLE INDEX
C                               5TH DIM: SECOND METASTABLE INDEX
C OUTPUT : (L*4)  LQCDA(,,) = .TRUE.  => QCD COEFFICIENT AVAILABLE
C                               .FALSE. => QDC COEFFICIENT NOT AVAILABLE
C                               1ST DIM: CHARGE STATE INDEX
C                               2ND DIM: FIRST METASTABLE INDEX
C                               3RD DIM: SECOND METASTABLE INDEX
C OUTPUT : (R*8)  XCDA(,,,,)= INTERPOLATION OF XCD COEFFICIENT (CM3 S-1)
C                               1ST DIM: TEMPERATURE INDEX
C                               2ND DIM: DENSITY INDEX
C                               3RD DIM: CHARGE STATE INDEX
C                               4TH DIM: FIRST PARENT METASTABLE INDEX
C                               5TH DIM: SECOND PARENT METASTABLE INDEX
C OUTPUT : (L*4)  LXCDA(,,) = .TRUE.  => XCD COEFFICIENT AVAILABLE
C                               .FALSE. => XDC COEFFICIENT NOT AVAILABLE
C                               1ST DIM: CHARGE STATE INDEX
C                               2ND DIM: FIRST PARENT METASTABLE INDEX
C                               3RD DIM: SECOND PARENT METASTABLE INDEX
C OUTPUT : (R*8)  PLTA(,,,) = INTERPOLATION OF PLT COEFFICIENT (W CM3 )
C                               1ST DIM: TEMPERATURE INDEX
C                               2ND DIM: DENSITY INDEX
```

```
C                              3RD DIM: CHARGE STATE INDEX
C                              4TH DIM: METASTABLE INDEX
C OUTPUT : (L*4)  LPLTA(,)   = .TRUE.  => PLT COEFFICIENT AVAILABLE
C                              .FALSE. => PLT COEFFICIENT NOT AVAILABLE
C                              1ST DIM: CHARGE STATE INDEX
C                              2ND DIM:  METASTABLE INDEX
C
C PROGRAM: (I*4)  IT         = GENERAL INDEX FOR TEMPERATURE
C          (I*4)  ID         = GENERAL INDEX FOR DENSITY
C          (I*4)  IZ         = GENERAL INDEX FOR CHARGE
C          (I*4)  IZ1        = GENERAL INDEX FOR CHARGE+1
C          (I*4)  IPRT       = GENERAL INDEX FOR PARENT METASTABLE
C          (I*4)  JPRT       = GENERAL INDEX FOR PARENT METASTABLE
C          (I*4)  IGRD       = GENERAL INDEX FOR METASTABLE
C          (I*4)  JGRD       = GENERAL INDEX FOR METASTABLE
C
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C
C
C AUTHOR: Alessandro Lanzafame, University of Strathclyde
C
C DATE:   21 October 1996
C
C-----------------------------------------------------------------------
C
C VERSION: 1.1                                          DATE: 12-03-98
C MODIFIED: RICHARD MARTIN
C               - PUT UNDER SCCS CONTROL
C
C-----------------------------------------------------------------------
      CHARACTER*120       DSFLLA(8)
      INTEGER             IDMAX,       IPDIMD,      ISDIMD,      ITDIMD
      INTEGER             ITMAX,       IZ0,         IZ1MAX,      IZ1MIN
      INTEGER             IZDIMD,      NDDIM,       NPART(IZDIMD)
      INTEGER             NPARTR(IZDIMD),           NTDIM
      LOGICAL             LACDA(IZDIMD,IPDIMD,IPDIMD)
      LOGICAL             LCCDA(IZDIMD,IPDIMD,IPDIMD)
      LOGICAL             LDEFA(8),    LEXSA(8),    LPART
      LOGICAL             LPLTA(IZDIMD,IPDIMD),     LPRBA(IZDIMD,IPDIMD)
      LOGICAL             LPRCA(IZDIMD,IPDIMD)
      LOGICAL             LQCDA(IZDIMD,IPDIMD,IPDIMD)
      LOGICAL             LSCDA(IZDIMD,IPDIMD,IPDIMD)
      LOGICAL             LSELA(8),    LXCDA(IZDIMD,IPDIMD,IPDIMD)
      REAL*8              ACDA(NTDIM,NDDIM,IZDIMD,IPDIMD,IPDIMD)
      REAL*8              CCDA(NTDIM,NDDIM,IZDIMD,IPDIMD,IPDIMD)
      REAL*8              DDENS(IDMAX),             DDENSD(ITDIMD)
      REAL*8              DRCOFD(ISDIMD,ITDIMD,ITDIMD)
      REAL*8              DRCOFI(ITMAX,IDMAX),      DTEV(ITMAX)
      REAL*8              DTEVD(ITDIMD)
      REAL*8              PLTA(NTDIM,NDDIM,IZDIMD,IPDIMD)
      REAL*8              PRBA(NTDIM,NDDIM,IZDIMD,IPDIMD)
      REAL*8              PRCA(NTDIM,NDDIM,IZDIMD,IPDIMD)
      REAL*8              QCDA(NTDIM,NDDIM,IZDIMD,IPDIMD,IPDIMD)
      REAL*8              SCDA(NTDIM,NDDIM,IZDIMD,IPDIMD,IPDIMD)
      REAL*8              XCDA(NTDIM,NDDIM,IZDIMD,IPDIMD,IPDIMD)
      REAL*8              ZDATA(ISDIMD)
```

5.60 d9mpop: Subroutine d9mpop from library adas4xx

```
C Copyright (c) 1995, Strathclyde University.
      SUBROUTINE D9MPOP( NTDIM , NDDIM , IZDIMD, IPDIMD,
     &                   NSTAGE, ITMAX , IDMAX , NPRT  , NMSUM ,
     &                   ACDA  , SCDA  , CCDA  , QCDA  , XCDA  ,
     &                   DENS  , DENSH ,
     &                   ITEM  , IDEN  ,
     &                   CFREC , CFION , CFMET ,
     &                   POPN  , POPNMO, POPNPO,
     &                   CPOPN , CPOPND, CPOPNZ,
     &                   POPF  ,
     &                   XTEMP , YTEMP , YTEM  ,
     &                   RHS   , RDUM  , SOLVE , LSOLVE
     &                  )
      IMPLICIT REAL*8(A-H,O-Z)
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN 77 SUBROUTINE: D9MPOP *******************
C
C  PURPOSE: CALCULATION OF METASTABLE RESOLVED IONISATION STAGE
C           POPULATIONS OF A PARTICULAR ELEMENT FOR A GIVEN TEMPERATURE
C           AND DENSITY. EXTENSION TO THE 2D (TEMPERATURE, DENSITY) CASE.
C
C  CALLING PROGRAM: ADAS409
C
C  SUBROUTINE:
C
C
C INPUT :(I*4) NTDIM         = MAXIMUM NUMBER OF TEMPERATURE VALUES
C INPUT :(I*4) NDDIM         = MAXIMUM NUMBER OF DENSITY VALUES
C INPUT :(I*4) IZDIMD        = MAXIMUM NUMBER OF STAGES-1
C INPUT :(I*4) IPDIMD        = MAXIMUM NUMBER OF METASTABLES FOR EACH
C                              IONISATION STAGE
C INPUT :(I*4) NSTAGE        = NUMBER OF STAGES-1
C INPUT :(I*4) ITMAX         = NUMBER OF TEMPERATURE VALUES
C INPUT :(I*4) IDMAX         = NUMBER OF DENSITY VALUES
C INPUT :(R*8) NPRT( )       = PARTITION OF TOTAL METASTABLES ACCORDING
C                              TO IONISATION STAGES
C                              1ST DIM: STAGE INDEX
C INPUT :(I*4) NMSUM         = TOTAL NUMBER OF POPULATIONS
C INPUT :(R*8) ACDA(,,,,)    = GENERALISED CR RECOMBINATION COEFFICIENT
C                              1ST DIM: TEMPERATURE INDEX
C                              2ND DIM: DENSITY INDEX
C                              3RD DIM: STAGE INDEX (LESS 1)
C                              4TH DIM: METASTABLE INDEX
C                              5TH DIM: METASTABLE INDEX
C INPUT :(R*8) SCDA(,,,,)    = GENERALISED CR IONISATION COEFFICIENT
C                              1ST DIM: TEMPERATURE INDEX
C                              2ND DIM: DENSITY INDEX
C                              3RD DIM: STAGE INDEX (LESS 1)
C                              4TH DIM: METASTABLE INDEX
C                              5TH DIM: METASTABLE INDEX
C INPUT :(R*8) CCDA(,,,,)    = GENERALISED CR CHARGE EXCH. COEFFICIENT
C                              1ST DIM: TEMPERATURE INDEX
C                              2ND DIM: DENSITY INDEX
C                              3RD DIM: STAGE INDEX (LESS 1)
C                              4TH DIM: METASTABLE INDEX
C                              5TH DIM: METASTABLE INDEX
C INPUT :(R*8) QCDA(,,,)     = GENERALISED CR CROSS-COUPL. COEFFICIENT
```

```
C                               1ST DIM: TEMPERATURE INDEX
C                               2ND DIM: DENSITY INDEX
C                               3RD DIM: STAGE INDEX (LESS 1)
C                               4TH DIM: METASTABLE INDEX
C                               5TH DIM: METASTABLE INDEX
C INPUT :(R*8) XCDA(,,,)      = GENERALISED CR PARENT X-CP. COEFFICIENT
C                               1ST DIM: TEMPERATURE INDEX
C                               2ND DIM: DENSITY INDEX
C                               3RD DIM: STAGE INDEX (LESS 1)
C                               4TH DIM: METASTABLE INDEX
C                               5TH DIM: METASTABLE INDEX
C INPUT :(R*8) DENS()         = ELECTRON DENSITIES FOR MODEL
C INPUT :(R*8) DENSH()        = NEUTRAL HYDROGEN DENSITIES FOR MODEL
C
C INPUT :(I*4) ITEM           = CURRENT TEMPERATURE INDEX
C INPUT :(I*4) IDEN           = CURRENT DENSITY INDEX
C
C OUTPUT:(R*8) CFREC( , , )   = RECOMBINATION RATE COEFFICIENTS TO ALL
C                               METASTABLE IPDIMD;STARTING FROM FIRST TO
C                               GROUND LEVEL,WITH CFREC(1,IPDIMD,IPDIMD)
C                               SET TO ZERO
C                               DIMENSIONS = (IPDIMD,IPDIMD,IZDIMD)
C OUTPUT:(R*8) CFION( , , )   = IONISATION RATE COEFFICIENTS TO ALL
C                               METASTABLE IPDIMD;STARTING FROM GROUND
C                               TO FIRST LEVEL,WITH
C                               CFION(NSTAGE,IPDIMD,IPDIMD)
C                               SET TO ZERO
C                               DIMENSIONS = (IPDIMD,IPDIMD,IZDIMD)
C OUTPUT:(R*8) CFMET( , , )   = CROSS COUPLING COEFFICIENTS BETWEEN
C                               METASTABLE IPDIMD WITH LEADING DIAGONAL
C                               CALCULATED
C                               DIMENSIONS = (IPDIMD,IPDIMD,IZDIMD)
C
C OUTPUT:(R*8) POPN( , , )    = ARRAY HOLDING POPULATION STATE VALUES
C                               WITH SECOND DIMENSION SET TO 1
C                               DIMENSIONS = (IPDIMD,NDONE,IZDIMD+1)
C OUTPUT:(R*8) POPNMO( , , )  = TEMPORARY NAME OF MATRIX HOLDING POPULATION
C                               STATE VALUES AFTER NORMALIZATION,TO BE
C                               SUBSTITUTED INTO NEXT EQUATION IN
C                               DOWNWARD LOOP
C                               DIMENSIONS = (IPDIMD,NDONE,IZDIMD+1)
C OUTPUT:(R*8) POPNPO( , , )  = TEMPORARY NAME OF MATRIX HOLDING POPULATION
C                               STATE VALUES AFTER NORMALIZATION,TO BE
C                               SUBSTITUTED INTO NEXT EQUATION IN UPWARD
C                               LOOP
C                               DIMENSIONS = (IPDIMD,NDONE,IZDIMD+1)
C
C OUTPUT:(R*8) CPOPN( , , )   = ARRAY HOLDING COEFFICIENTS OF POPULATION
C                               STATE EQUATIONS
C                               DIMENSIONS = (IPDIMD,IPDIMD,IZDIMD+1)
C OUTPUT:(R*8) CPOPND( , , )  = TEMPORARY NAME OF MATRIX TO BE SUBSTITUTED
C                               INTO NEXT EQUATION IN UPWARD LOOP
C                               DIMENSIONS = (IPDIMD,IPDIMD,IZDIMD+1)
C OUTPUT:(R*8) CPOPNZ( , , )  = TEMPORARY NAME OF MATRIX TO BE SUBSTITUTED
C                               INTO NEXT EQUATION  IN DOWNWARD LOOP
C                               DIMENSIONS = (IPDIMD,IPDIMD,IZDIMD+1)
C
C OUTPUT:(R*8) POPF()         = POPULATIONS FOR A SPECIFIED TEMPERATURE  D
C                               1ST DIM: INDEX OVER STAGES/METASTABLES
C
C OUTPUT:(R*8) XTEMP( , )     = TEMPORARY MATRIX USED DURING SUBROUTINE
```

```
C                               CALCULATIONS
C                               DIMENSIONS = (IPDIMD,IPDIMD)
C OUTPUT:(R*8) YTEMP( , )     = TEMPORARY MATRIX FOR DURING SUBROUTINE
C                               CALCULATIONS
C                               DIMENSIONS = (IPDIMD,IPDIMD)
C OUTPUT:(R*8) YTEM( )        = TEMPORARY ARRAY FOR HOLDING VALUES OF
C                               DIFFERENCE BETWEEN RECOMBINATION AND
C                               IONISATION GROUND LEVEL COEFFICIENTS
C                               DIMENSIONS = (NSTAGE)
C
C OUTPUT:(R*8) RHS( )         = SIPHONED OFF COLUMN OF NORMALIZATION
C                               MATRIX,USED TO CALCULATE METASTABLE
C                               IPDIMD OF DOMINANT STAGE THROUGH MATINV
C                               DIMENSIONS = (2*IPDIMD-1)
C OUTPUT:(R*8) RDUM( )        = DUMMY ARRAY USED IN MATINV AS RHS WHEN
C                               LSOLVE =  FALSE
C OUTPUT:(R*8) SOLVE( , )     = NORMALIZATION MATRIX AT CRITICAL STAGE
C                               DIMENSIONS = (2*IPDIMD-1,2*IPDIMD-1)
C OUTPUT:(L*4) LSOLVE         = .TRUE.  => SOLVE SET OF EQUATIONS
C                             = .FALSE. => INVERT MATRIX ONLY
C
C        (I*4) NDONE          = PARAMETER = 1 TO ALLOW 3D MATRIX USE
C        (I*4) ID             = POSITION OF DOMINANT TERM
C        (I*4) ISTATE         = STAGE INDEX
C        (I*4) ITEM           = GENERAL INDEX
C        (I*4) I              = GENERAL INDEX
C        (I*4) J              = GENERAL INDEX
C        (I*4) K              = GENERAL INDEX
C        (R*8) YMIN           = VALUE OF DIFFERENCE BETWEEN
C                               RECOMBINATION AND IONISATION COEFFICIENTS
C                               OF GROUND IPDIMD
C
C  ROUTINES:
C           ROUTINE    SOURCE    BRIEF DESCRIPTION
C           ------------------------------------------------------------
C           D5DIAG     ADAS      SETS UP ON-DIAGONAL ELEMENT OF MATRIX
C           D5MFSP     ADAS      EXECUTES PARTITION MATRIX INVERSION
C           DXMADD     ADAS      MATRIX ADDITION/SUBTRACTION
C           DXMMUL     ADAS      MATRIX MULTIPLICATION
C           XXMINV     ADAS      MATRIX INVERSION
C
C
C  AUTHOR:  Alessandro Lanzafame, University of Strathclyde
C
C  DATE:    11 December 1995
C
C-----------------------------------------------------------------------
C
C VERSION: 1.1                                          DATE: 12-03-98
C MODIFIED: RICHARD MARTIN
C               - PUT UNDER SCCS CONTROL
C
C-----------------------------------------------------------------------
      INTEGER             IDEN,        IDMAX,       IPDIMD,      ITEM
      INTEGER             ITMAX,       IZDIMD,      NDDIM,       NMSUM
      INTEGER             NPRT(IZDIMD),             NSTAGE,      NTDIM
      LOGICAL             LSOLVE
      REAL*8              ACDA(NTDIM,NDDIM,IZDIMD,IPDIMD,IPDIMD)
      REAL*8              CCDA(NTDIM,NDDIM,IZDIMD,IPDIMD,IPDIMD)
      REAL*8              CFION(IPDIMD,IPDIMD,IZDIMD)
      REAL*8              CFMET(IPDIMD,IPDIMD,IZDIMD)
```

```
REAL*8            CFREC(IPDIMD,IPDIMD,IZDIMD)
REAL*8            CPOPN(IPDIMD,IPDIMD,IZDIMD+1)
REAL*8            CPOPND(IPDIMD,IPDIMD,IZDIMD+1)
REAL*8            CPOPNZ(IPDIMD,IPDIMD,IZDIMD+1)
REAL*8            DENS(NDDIM), DENSH(NTDIM)
REAL*8            POPF(NMSUM), POPN(IPDIMD,NDONE,IZDIMD+1)
REAL*8            POPNMO(IPDIMD,NDONE,IZDIMD+1)
REAL*8            POPNPO(IPDIMD,NDONE,IZDIMD+1)
REAL*8            QCDA(NTDIM,NDDIM,IZDIMD,IPDIMD,IPDIMD)
REAL*8            RDUM(IPDIMD),             RHS(2*IPDIMD-1)
REAL*8            SCDA(NTDIM,NDDIM,IZDIMD,IPDIMD,IPDIMD)
REAL*8            SOLVE(2*IPDIMD-1,2*IPDIMD-1)
REAL*8            XCDA(NTDIM,NDDIM,IZDIMD,IPDIMD,IPDIMD)
REAL*8            XTEMP(IPDIMD,IPDIMD),     YTEM(IZDIMD)
REAL*8            YTEMP(IPDIMD,IPDIMD)
```

5.61 d9rdnm: Subroutine d9rdnm from library adas4xx

```
C Copyright (c) 1997, Strathclyde University.
      SUBROUTINE D9RDNM( DSNINC , LPART  , IFAIL  ,
     &                   IZ0    , NPART  , IPRTD  , IGRDD , ICLASS ,
     &                   IZ1    , ITMAX  , IDMAX  ,
     &                   ISDIMD , IZDIMD , ITDIMD ,
     &                   ISMAXD , IZMAXD , ITMAXD , IDMAXD , NPARTR,
     &                   DTEV   , DDENS  ,
     &                   DTEVD  , DDENSD , DRCOFD , ZDATA  ,
     &                   DRCOFI
     &                 )
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: D9RDNM *********************
C
C PURPOSE : TO EXTRACT COLLISIONAL DIELECTRONIC DATA FOR A
C           (TEMPERATURE, DENSITY) GRID FROM
C           EITHER PARTIAL (METASTABLE/PARENT RESOLVED) OR STANDARD
C           (UNRESOLVED) ISONUCLEAR MASTER FILES
C
C NOTE    : THE SOURCE DATA IS CONTAINED AS SEQUENTIAL DATASETS
C           WITH THE FOLLOWING NAMING CONVENTIONS:
C
C                   (1) JETSHP.ACD<YR>#<EL).<CODE>DATA
C                   (2) JETSHP.SCD<YR>#<EL>.<CODE>DATA
C                   (3) JETSHP.CCD<YR>#<EL>.<CODE>DATA
C                   (4) JETSHP.PRB<YR>#<EL>.<FILT>.<CODE>DATA
C                   (5) JETSHP.PRC<YR>#<EL>.<FILT>.<CODE>DATA
C                   (6) JETSHP.QCD<YR>#<EL>.<CODE>DATA
C                   (7) JETSHP.XCD<YR>#<EL>.<CODE>DATA
C                   (8) JETSHP.PLT<YR>#<EL>.<CODE>DATA
C                   (9) JETSHP.PLS<YR>#<EL>.<CODE>DATA
C
C       WHERE, <YR>   = TWO DIGIT YEAR NUMBER
C              <EL>   = ONE OR TWO CHARACTER ELEMENT SYMBOL
C              <CODE> = R       => PARTIAL DATA
C                       U       => PARTIAL DATA
C                       OMITTED => STANDARD DATA
C              <FILT> = SIX CHARACTER POWER FILTER CODE
C
C       AND DATA OF CLASSES 6 AND 7 DO NOT EXIST FOR THE PARTIAL CASE.
C
C
C INPUT  : (C*120) DSNINC   = ISONUCLEAR MASTER FILE NAME - VERIFIED
C                             AND READY FOR DYNAMIC ALLOCATION.
C INPUT  : (L*4)  LPART     = .TRUE.  => PARTIAL (RESOLVED) MASTER DATA
C                           . FALSE. => UNSRESOLVED MASTER DATA
C INPUT  : (I*4)  IZ0       = NUCLEAR CHARGE
C INPUT  : (I*4)  NPART()   = METASTABLE PARTITION.  I.E. NUMBER OF
C                             METASTABLES FROM CHARGE STATE IZ1MIN-1 TO
C                             IZ1MAX ON INPUT
C INPUT  : (I*4)  IPRTD     = REQUIRED PARENT INDEX
C INPUT  : (I*4)  IGRDD     = REQUIRED GROUND INDEX
C INPUT  : (I*4)  ICLASS    = CLASS OF DATA (1 - 9 )
C INPUT  : (I*4)  IZ1       = REQUIRED ION CHARGE + 1
C INPUT  : (I*4)  ITMAX     = NUMBER OF DTEV() VALUES
C INPUT  : (I*4)  IDMAX     = NUMBER OF DDENS() VALUES
C INPUT  : (I*4)  ISDIMD    = MAXIMUM NUMBER OF (CHARGE, PARENT, GROUND)
C                             BLOCKS IN ISONUCLEAR MASTER FILES
```

```
C INPUT  : (I*4)  IZDIMD    = MAXIMUM NUMBER OF CHARGE STATES
C                             IN ISONUCLEAR MASTER FILES
C INPUT  : (I*4)  ITDIMD    = MAXIMUM NUMBER OF TEMP OR DENS VALUES IN
C                             ISONUCLEAR MASTER FILES
C INPUT  : (R*8)  DTEV()    = DLOG10(ELECTRON TEMPERATURES (EV))
C INPUT  : (R*8)  DDENS()   = DLOG10(ELECTRON DENSITIES (CM-3))
C
C OUTPUT : (I*4)  IFAIL     = 0    IF ROUTINE SUCCESSFUL - DATA FOR THE
C                                  REQUESTED YEAR USED.
C                             = 1    IF ROUTINE OPEN STATEMENT FAILED
C                             = 2    IF FILE EXISTS BUT REQUIRED DATA
C                                  BLOCK DOES NOT
C OUTPUT : (I*4)  ISMAXD    = NUMBER OF (CHARGE, PARENT, METASTABLE)
C                             BLOCKS IN SELECTED MASTER FILE
C OUTPUT : (I*4)  IZMAXD    = NUMBER OF ZDATA() VALUES IN SELECTED
C                             MASTER FILE
C OUTPUT : (I*4)  ITMAXD    = NUMBER OF DTEVD() VALUES IN SELECTED
C                             MASTER FILE
C OUTPUT : (I*4)  IDMAXD    = NUMBER OF DDENSD() VALUES IN SELECTED
C                             MASTER FILE
C OUTPUT : (I*4)  NPARTR()  = METASTABLE PARTITION.  I.E. NUMBER OF
C                             METASTABLES FROM CHARGE STATE IZ1MIN-1 TO
C                             IZ1MAX FOUND IN MASTER FILE
C OUTPUT : (R*8)  DTEVD()   = DLOG10(DATA ELECTRON TEMPERATURES (EV))
C                             IN SELECTED MASTER FILE
C OUTPUT : (R*8)  DDENSD()  = DLOG10(DATA ELECTRON DENSITIES (CM-3))
C                             IN SELECTED MASTER FILE
C OUTPUT : (R*8)  DRCOFD(,,)= DLOG10(DATA RATE COEFFICIENTS (CM-3/S))
C                             IN SELECTED MASTER FILE
C                             1ST DIM: (CHARGE,META,GRD) BLOCK INDEX
C                             2ND DIM: TEMPERATURE INDEX
C                             3RD DIM: DENSITY INDEX
C OUTPUT : (R*8)  ZDATA()   = CHARGE + 1 FOR IONS IN SELECTED MASTER
C                             FILE
C                             1ST DIM: (CHARGE,META,GRD) BLOCK INDEX
C OUTPUT : (R*8)  DRCOFI(,) = INTERPOLATION OF DRCOFD(,,) FOR
C                             DTEV() & DDENS()
C
C PROGRAM: (C*80) DSNOLD    = FILE NAME USED IN PREVIOUS CALL
C          (C*80) CLINE     = GENERAL CHARACTER VARIABLE
C          (C*80) CTERM     = TERMINATOR LINE - '-' FILLED VARIABLE
C          (C*4)) CPATRN()  = PATTERN USED TO DETECT DATA CLASS
C          (I*4)  IZ0D      = NUCLEAR CHARGE READ FROM MASTER FILE
C          (I*4)  IZ1MIN    = MINIMUM CHARGE+1 READ FROM MASTER FILE
C          (I*4)  IZ1MAX    = MAXIMUM CHARGE+1 READ FROM MASTER FILE
C          (I*4)  IABT      = ABORT CODE
C          (I*4)  INDSEL    = LOCATION OF (CHARGE,PRNT,GRND)
C                             DATA BLOCK IN FILE
C          (I*4)  IZDAT     = CURRENT DATA BLOCK ION CHARGE +1
C          (I*4)  ISEL      = GENERAL INDEX
C          (I*4)  I         = GENERAL INDEX
C          (I*4)  IT        = GENERAL INDEX
C          (I*4)  ID        = GENERAL INDEX
C          (I*4)  IZCHK     = INDEX TO VERIFY DATA Z1 SET COMPLETE
C          (I*4)  IPRTR()   = PARENT INDICES IN DATA SET
C          (I*4)  IGRDR()   = GROUND INDICES IN DATA SET
C          (I*4)  LCK       = MUST BE GREATER THAN 'ITMAXD' & 'IDMAXD'
C                             & 'ITMAX' - ARRAY SIZE FOR SPLINE CALCS.
C          (R*8)  A()       = GENERAL ARRAY
C          (R*8)  DRCOF0(,) = INTERPOLATION OF DRCOFD(,,) W.R.T DTEV()
C          (L*8)  LEXIST    = TRUE --- FILE TO OPEN EXISTS ELSE NOT
```

```
C          (I*4)  L1      = PARAMETER = 1
C          (I*4)  IOPT    = DEFINES THE BOUNDARY DERIVATIVES FOR THE
C                              SPLINE ROUTINE 'XXSPLN', SEE 'XXSPLN'.
C          (L*4)  LSETX   = .TRUE.  => SET UP SPLINE PARAMETERS RELATING
C                                      TO X-AXIS.
C                           .FALSE. => DO NOT SET UP SPLINE PARAMETERS
C                                      RELATING TO X-AXIS.
C                                      (I.E. THEY WERE SET IN A PREVIOUS
C                                            CALL )
C                           (VALUE SET TO .FALSE. BY 'XXSPLN')
C          (R*8)  DY()    = SPLINE INTERPOLATED DERIVATIVES
C
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4UNIT     ADAS      FETCH UNIT NUMBER FOR OUTPUT OF MESSAGES
C          I4FCTN     ADAS      CONVERT STRING TO INTEGER FORM
C
C          (R*8 ADAS FUNCTION - 'R8FUN1' ( X -> X) )
C
C AUTHOR : Alessandro Lanzafame
C
C DATE   : 8th December 1995
C
C-----------------------------------------------------------------------
C
C VERSION: 1.1                                          DATE: 12-03-98
C MODIFIED: RICHARD MARTIN
C               - PUT UNDER SCCS CONTROL
C
C-----------------------------------------------------------------------
      CHARACTER*120       DSNINC
      INTEGER             ICLASS,      IDMAX,       IDMAXD,      IFAIL
      INTEGER             IGRDD,       IPRTD,       ISDIMD,      ISMAXD
      INTEGER             ITDIMD,      ITMAX,       ITMAXD,      IZ0
      INTEGER             IZ1,         IZDIMD,      IZMAXD
      INTEGER             NPART(IZDIMD),            NPARTR(IZDIMD)
      LOGICAL             LPART
      REAL*8              DDENS(IDMAX),             DDENSD(ITDIMD)
      REAL*8              DRCOFD(ISDIMD,ITDIMD,ITDIMD)
      REAL*8              DRCOFI(ITMAX,IDMAX),      DTEV(ITMAX)
      REAL*8              DTEVD(ITDIMD),            ZDATA(ISDIMD)
```

5.62 d9scrp: Subroutine d9scrp from library adas4xx

```
       SUBROUTINE D9SCRP( LRSCRP , LSNULL  ,
     &                    DSNINC , DSPECA  ,
     &                    NDLINE , NDCOMP  , NDRAT  , NDFILE ,
     &                    NFILE  , LFILE   ,
     &                    UID    , GROUP   , TYPE   , EXT    , ION    ,
     &                    MEMB   , IZ0     ,
     &                    NLINE  , NCOMP   ,
     &                    IZION  , IMET    , CIMET  , INDPH  , CINDPH ,
     &                    IFILE  , TITL    ,
     &                    NRAT   ,
     &                    ILINE  , JLINE   , TITR   , IRCODE
     &                  )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: D9SCRP  *********************
C
C  PURPOSE: TO READ SCRIPT FILE AND ACCESS EMISSIVITY DATA
C           ON SPECTRAL LINES REQUESTED FOR FURTHER PROCESSING IN
C           EQUILIBRIUM IONISATION CODES.
C
C  CALLING PROGRAM: ADAS409
C
C  SUBROUTINE:
C
C  INPUT : (C*120) DSNINC   = SCRIPT DATA SET NAME (FULL MVS DSN)
C                             (IN FORM SUITABLE FOR DYNAMIC ALLOCATION)
C  INPUT : (I*4)   NDLINE   = MAXIMUM NUMBER OF LINES ALLOWED
C  INPUT : (I*4)   NDCOMP   = MAXIMUM NUMBER OF COMPONENT FOR EACH LINE
C  INPUT : (I*4)   NDRAT    = MAXIMUM NUMBER OF LINE RATIOS ALLOWED
C  INPUT : (I*4)   NDFILE   = MAXIMUM NUMBER OF EMISSIVITY FILES WHICH
C                             CAN BE SEARCHED
C
C  OUTPUT: (L*4)   LRSCRP   = .TRUE.  => SCRIPT FILE READ
C                             .FALSE. => SCRIPT FILE NOT READ
C  OUTPUT: (L*4)   LSNULL   = .TRUE.  => SCRIPT FILE SET TO NULL
C                             .FALSE. => SCRIPT FILE VALID
C  OUTPUT: (C*120) DSPECA() = PHOTON EMISSIVITY SOURCE FILES
C  OUTPUT: (I*4)   NFILE    = NUMBER OF PEC FILES TO BE SCANNED
C  OUTPUT: (L*4)   LFILE()  = .TRUE.  => PEC FILE EXISTS AND MATCHES
C                             .FALSE. => PEC FILE DOES NOT EXIST/MATCH
C  OUTPUT: (C*6)   UID()    = USER IDENTIFIER OF PEC FILE
C  OUTPUT: (C*8)   GROUP()  = GROUP IDENTIFIER OF PEC FILE
C  OUTPUT: (C*5)   TYPE()   = TYPE IDENTIFIER OF PEC FILE
C  OUTPUT: (C*3)   EXT()    = EXTENSION OF PEC FILE MEMBER NAME
C  OUTPUT: (C*4)   ION()    = ION NAME OF PEC FILE MEMBER NAME
C  OUTPUT: (C*8)   MEMB()   = MEMBER NAME OF PEC FILE
C  OUTPUT: (I*4)   NLINE    = NUMBER OF LINES IDENTIFIED IN SCRIPT
C  OUTPUT: (I*4)   NCOMP()  = NUMBER OF COMPONENTS OF SCRIPT LINE
C                             1ST DIM: LINE INDEX
C  OUTPUT: (I*4)   IZION(,) = CHARGE STATE OF COMPONENT
C                             1ST DIM: LINE INDEX
C                             2ND DIM: COMPONENT INDEX
C  OUTPUT: (I*4)   IMET(,)  = NUMBER OF COMPONENTS OF SCRIPT LINE
C                             1ST DIM: LINE INDEX
C                             2ND DIM: COMPONENT INDEX
C  OUTPUT: (C*1)   CIMET(,) = SIGN (+, BLANK OR -) OF METASTABLE
C                             1ST DIM: LINE INDEX
C                             2ND DIM: COMPONENT INDEX
C  OUTPUT: (I*4)   INDPH(,) = PEC FILE INDEX OF LINE COMPONENT
```

```
C                              1ST DIM: LINE INDEX
C                              2ND DIM: COMPONENT INDEX
C  OUTPUT: (C*1)   CINDPH(,)= DRIVER (E OR BLANK => ELECTRONS)
C                                    (H          => HYDROGEN )
C                              1ST DIM: LINE INDEX
C                              2ND DIM: COMPONENT INDEX
C  OUTPUT: (I*4)   IFILE(,) = INDEX OF PEC FILE IN FILE LIST
C                              1ST DIM: LINE INDEX
C                              2ND DIM: COMPONENT INDEX
C  OUTPUT: (C*12)  TITL(,)  = TITLE FOR LINE COMPONENT
C                              1ST DIM: LINE INDEX
C                              2ND DIM: COMPONENT INDEX
C  OUTPUT: (I*4)   NRAT     = NUMBER OF RATIOS IDENTIFIED IN SCRIPT
C  OUTPUT: (I*4)   ILINE()  = INDEX OF NUMERATOR LINE FOR LINE RATIO
C  OUTPUT: (I*4)   JLINE()  = INDEX OF DENOMINATOR LINE FOR LINE RATIO
C  OUTPUT: (C*25)  TITR()   = TILE FOR LINE RATIO
C  OUTPUT: (I*4)   IRCODE   = ERROR FLAG:
C                              0 => SCRIPT FILE WAS READ OKAY
C                              1 => SCRIPT FILE DOES NOT EXIST
C                              2 => I/O ERROR READING THE SCRIPT FILE
C                              3 => 1 OR MORE FILE NAMES IN SCRIPT FILE
C                                   IS/ARE INVALID.
C
C          (I*4)   IUNT10  = PARAMETER = INPUT UNIT FOR DATA
C          (L*4)   OPEN10  = .TRUE.  => FILE ALLOCATED TO UNIT 10.
C                            .FALSE. => NO FILE ALLOCATED TO UNIT 10.
C
C ROUTINES:
C          ROUTINE    SOURCE   BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          XXSLEN     ADAS      FIND NON-BLANK CHARACTERS IN STRING
C
C
C AUTHOR:  H. P. SUMMERS, JET (ORIGINALLY D5SCRP.FOR)
C          K1/1/57
C          JET EXT. 4941
C
C DATE:    20/04/94
C
C UPDATE:  ALESSANDRO LANZAFAME, CONVERTED TO D9SCRP.FOR
C
C-----------------------------------------------------------------------
C
C VERSION: 1.1                                          DATE: 12-03-98
C MODIFIED: RICHARD MARTIN
C               - PUT UNDER SCCS CONTROL
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      CHARACTER           CIMET(NDLINE,NDCOMP)
      CHARACTER           CINDPH(NDLINE,NDCOMP)
      CHARACTER*120       DSNINC,      DSPECA(NDFILE)
      CHARACTER*3         EXT(NDFILE)
      CHARACTER*8         GROUP(NDFILE)
      CHARACTER*4         ION(NDFILE)
      CHARACTER*8         MEMB(NDFILE)
      CHARACTER*12        TITL(NDLINE,NDCOMP)
      CHARACTER*25        TITR(NDRAT)
      CHARACTER*5         TYPE(NDFILE)
      CHARACTER*6         UID(NDFILE)
```

```
INTEGER            IFILE(NDLINE,NDCOMP),     ILINE(NDRAT)
INTEGER            IMET(NDLINE,NDCOMP),      INDPH(NDLINE,NDCOMP)
INTEGER            IRCODE,      IZ0,         IZION(NDLINE,NDCOMP)
INTEGER            JLINE(NDRAT),             NCOMP(NDLINE)
INTEGER            NDCOMP,      NDFILE,      NDLINE,      NDRAT
INTEGER            NFILE,       NLINE,       NRAT
LOGICAL            LFILE(NDFILE),            LRSCRP,      LSNULL
```

5.63 d9sgcf: Subroutine d9sgcf from library adas4xx

```
C Copyright (c) 1997, Strathclyde University.
      SUBROUTINE D9SGCF( IZ0    , IZL    , IZH    ,
     &                   ISDIMD , IZDIMD , ITDIMD , IPDIMD , IMDIMD ,
     &                   NMSUM  , IZIP   , IMIP   , IPIZM  ,
     &                   NDLINE , NDCOMP ,
     &                   NLINE  , NCOMP  , SPECL  , IPLINE ,
     &                   IZION  , IMET   , CIMET  , INDPH  , CINDPH ,
     &                   IFILE  ,
     &                   NTDIM  , NDDIM  , ITMAX  , IDMAX  ,
     &                   DENS   , DENSH  ,
     &                   PECA   , LPEC   ,
     &                   FPABUN ,
     &                   GCFPEQ , GCFEQ  ,
     &                   NDRAT  , NRAT   ,
     &                   ILINE  , JLINE  ,
     &                   RATA
     &                  )
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: D9SGCF *********************
C
C PURPOSE : TO ASSEMBLE GCF FUNCTIONS AND THEIR COMPONENTS USING
C           FRACTIONAL METASTABLE ABUNDANCES. 2D (TEMPERATURE, DENSITY)
C           VERSION.
C
C
C
C INPUT  : (I*4)  IZ0       = NUCLEAR CHARGE
C INPUT  : (I*4)  IZL       = MINIMUM ION CHARGE+1 IN MASTER DATA FILES
C INPUT  : (I*4)  IZH       = MAXIMUM ION CHARGE+1 IN MASTER DATA FILES
C INPUT  : (I*4)  ISDIMD    = MAXIMUM NUMBER OF (CHARGE, PARENT, GROUND)
C                             BLOCKS IN ISONUCLEAR MASTER FILES
C INPUT  : (I*4)  IZDIMD    = MAXIMUM NUMBER OF CHARGE STATES
C                             IN ISONUCLEAR MASTER FILES
C INPUT  : (I*4)  ITDIMD    = MAXIMUM NUMBER OF TEMP OR DENS VALUES IN
C                             ISOELECTRONIC MASTER FILES
C INPUT  : (I*4)  IPDIMD    = MAXIMUM NUMBER OF METASTABLES FOR EACH
C                             IONISATION STAGE
C INPUT  : (I*4)  IMDIMD    = MAXIMUM NUMBER OF METASTABLES
C
C INPUT  : (I*4)  NMSUM     = TOTAL NUMBER OF POPULATIONS
C
C INPUT  :        IZIP()    = ION CHARGE +1 (IZ1) OF METASTABLE IN LIST
C INPUT  :        IMIP()    = METASTABLE INDEX WITHIN CHARGE STATE IZ1
C                             OF METASTABLE INDEX FROM COMPLETE LIST
C INPUT  :        IPIZM(,)  = METASTABLE INDEX IN COMPLETE LIST
C                             1ST DIM: INDEX IZ1-IZL+1
C                             2ND DIM: METASTABLE COUNT FOR STAGE (IGRD)
C INPUT  : (I*4)  NDLINE    = MAXIMUM NUMBER OF LINES ALLOWED
C INPUT  : (I*4)  NDCOMP    = MAXIMUM NUMBER OF COMPONENT FOR EACH LINE
C INPUT  : (I*4)  NLINE     = NUMBER OF LINES IDENTIFIED IN SCRIPT
C INPUT  : (I*4)  NCOMP()   = NUMBER OF COMPONENTS OF SCRIPT LINE
C INPUT  : (I*4)  IZION(,)  = CHARGE STATE OF COMPONENT
C                             1ST DIM: LINE INDEX
C                             2ND DIM: COMPONENT INDEX
C INPUT  : (I*4)  IMET(,)   = METASTABLE INDEX OF COMPONENT OF
C                             SCRIPT LINE WITHIN CHARGE STATE
C                             1ST DIM: LINE INDEX
```

```
C                                2ND DIM: COMPONENT INDEX
C INPUT  : (C*1)  CIMET(,)  = SIGN (+, BLANK OR -) OF METASTABLE
C                                1ST DIM: LINE INDEX
C                                2ND DIM: COMPONENT INDEX
C INPUT  : (I*4)  INDPH(,)  = PEC FILE INDEX OF LINE COMPONENT
C                                1ST DIM: LINE INDEX
C                                2ND DIM: COMPONENT INDEX
C INPUT  : (C*1   CINDPH(,) = DRIVER (E OR BLANK => ELECTRONS)
C                                       (H          => HYDROGEN )
C                                1ST DIM: LINE INDEX
C                                2ND DIM: COMPONENT INDEX
C INPUT  : (I*4)  IFILE(,)  = INDEX OF PEC FILE IN FILE LIST
C                                1ST DIM: LINE INDEX
C                                2ND DIM: COMPONENT INDEX
C INPUT  : (I*4)  NTDIM     = MAXIMUM NUMBER OF DTEV VALUES
C INPUT  : (I*4)  NDDIM     = MAXIMUM NUMBER OF DDENS VALUES
C INPUT  : (I*4)  ITMAX     = NUMBER OF DTEV() VALUES
C INPUT  : (I*4)  IDMAX     = NUMBER OF DDENS() VALUES
C INPUT  : (R*8)  DENS()    = ELECTRON DENSITIES (CM-3))
C INPUT  : (R*8)  DENSH()   = HYDROGEN DENSITIES (CM-3))
C INPUT  : (R*8)  PECA(,,,) = PHOTON EMISSIVITY COEFFICIENTS (CM3 S-1)
C                                1ST DIM: TEMPERATURE INDEX
C                                2ND DIM: DENSITY INDEX
C                                3RD DIM: LINE INDEX
C                                4TH DIM: COMPONENT INDEX
C INPUT  : (L*4)  LPEC(,)   = .TRUE.  => PHOTON EMISSIVITY OBTAINED
C                                .FALSE. => PHOTON EMISSIVITY NOT OBTAINED
C                                2ND DIM: LINE INDEX
C                                3RD DIM: COMPONENT INDEX
C INPUT  : (R*8)  FPABUN(,,)= RESOLVED METASTABLE EQUILIBRIUM
C                                FRACTIONAL ABUNDANCES
C                                1ST DIM: - TEMPERATURE INDEX
C                                2ND DIM: - DENSITY INDEX
C                                3RD DIM: - METASTABLE INDEX
C INPUT  : (I*4)  NDRAT     = MAXIMUM NUMBER OF LINE RATIOS ALLOWED
C INPUT  : (I*4)  NRAT      = NUMBER OF RATIOS IDENTIFIED IN SCRIPT
C INPUT  : (I*4)  ILINE()   = INDEX OF NUMERATOR LINE FOR LINE RATIO
C INPUT  : (I*4)  JLINE()   = INDEX OF DENOMINATOR LINE FOR LINE RATIO
C
C OUTPUT : (C*16) SPECL(,)  = SPEC. OF POINTERS OF LINE COMPONENT
C                                1ST DIM: LINE INDEX
C                                2ND DIM: COMPONENT INDEX
C OUTPUT : (I*4)  IPLINE(,) = METASTABLE POINTER OF LINE COMPONENT
C                                1ST DIM: LINE INDEX
C                                2ND DIM: COMPONENT INDEX
C OUTPUT : (R*8)  GCFPEQ(,,,)=GCF FUNC. COMPONENT (CM3 S-1)
C                                1ST DIM: TEMPERATURE INDEX
C                                2ND DIM: DENSITY INDEX
C                                3RD DIM: LINE INDEX
C                                4TH DIM: LINE COMPONENT INDEX
C OUTPUT : (R*8)  GCFEQ(,,) = GCF FUNCTION  (CM3 S-1)
C                                1ST DIM: TEMPERATURE INDEX
C                                2ND DIM: DENSITY INDEX
C                                3RD DIM: LINE INDEX
C OUTPUT : (R*8)  RATA(,,)  = LINE GCF RATIOS
C                                1ST IND: TEMPERATURE INDEX
C                                2ND IND: DENSITY INDEX
C                                3RD IND: RATIO INDEX
C
C
C PROGRAM: (I*4)  IT        = GENERAL INDEX FOR TEMPERATURE
```

```
C          (I*4)  ID        = GENERAL INDEX FOR DENSITY
C          (I*4)  IZ        = GENERAL INDEX FOR CHARGE
C          (I*4)  IP        = GENERAL INDEX FOR CHARGE
C          (I*4)  IZ1       = GENERAL INDEX FOR CHARGE+1
C          (I*4)  IL        = GENERAL INDEX FOR LINE
C          (I*4)  IR        = GENERAL INDEX FOR RATIO
C          (I*4)  ICPT      = GENERAL INDEX FOR LINE COMPONENT
C
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C
C
C AUTHOR : Alessandro Lanzafame, University of Strathclyde
C
C DATE   : 11 December 1995
C
C-----------------------------------------------------------------------
C
C VERSION: 1.1                                          DATE: 12-03-98
C MODIFIED: RICHARD MARTIN
C               - PUT UNDER SCCS CONTROL
C
C-----------------------------------------------------------------------
      CHARACTER           CIMET(NDLINE,NDCOMP)
      CHARACTER           CINDPH(NDLINE,NDCOMP)
      CHARACTER*16        SPECL(NDLINE,NDCOMP)
      INTEGER             IDMAX,       IFILE(NDLINE,NDCOMP)
      INTEGER             ILINE(NDRAT),             IMDIMD
      INTEGER             IMET(NDLINE,NDCOMP),      IMIP(IMDIMD)
      INTEGER             INDPH(NDLINE,NDCOMP),     IPDIMD
      INTEGER             IPIZM(IZDIMD,IPDIMD)
      INTEGER             IPLINE(NDLINE,NDCOMP),    ISDIMD,      ITDIMD
      INTEGER             ITMAX,       IZ0,         IZDIMD,      IZH
      INTEGER             IZION(NDLINE,NDCOMP),     IZIP(IMDIMD)
      INTEGER             IZL,         JLINE(NDRAT)
      INTEGER             NCOMP(NDLINE),            NDCOMP,      NDDIM
      INTEGER             NDLINE,      NDRAT,       NLINE,       NMSUM
      INTEGER             NRAT,        NTDIM
      LOGICAL             LPEC(NDLINE,NDCOMP)
      REAL*8              DENS(NDDIM), DENSH(NDDIM)
      REAL*8              FPABUN(NTDIM,NDDIM,IMDIMD)
      REAL*8              GCFEQ(NTDIM,NDDIM,NDLINE)
      REAL*8              GCFPEQ(NTDIM,NDDIM,NDLINE,NDCOMP)
      REAL*8              PECA(NTDIM,NDDIM,NDLINE,NDCOMP)
      REAL*8              RATA(NTDIM,NDDIM,NDRAT)
```

5.64 d9spc2: Subroutine d9spc2 from library adas4xx

```
C Copyright (c) 1997, Strathclyde University.
      SUBROUTINE D9SPC2( DSNAME, IBSEL  , IZIN   , IZ0IN  ,
     &                   ITVAL  , IDVAL , TVAL   , DVAL   ,
     &                   WLNGTH ,
     &                   PECA   , LTRNG  , LDRNG  ,
     &                   TITLX  , IRCODE
     &                 )
C-----------------------------------------------------------------------
C
C  ******************* FORTRAN77 SUBROUTINE: D9PSC2 *******************
C
C  PURPOSE: TO EXTRACT AND INTERPOLATE  PHOTON EMISSIVITIES FOR
C           EMITTING IONS.
C
C           DERIVED FROM D5SPC2
C
C           THIS ROUTINE TAKES AS INPUT THE NAMES OF THE PHOTON
C           EMISSIVITY FILES AND CHECKS THEY ARE THERE BEFORE
C           OPENING THEM AND EXTRACTING ALL REQUIRED INFORMATION.
C
C  CALLING PROGRAM: D9SPEC
C
C  SUBROUTINE:
C
C  INPUT : (I*4)   IBSEL   = INDEX OF DATA-BLOCK SELECTED FOR ANALYSIS
C  INPUT : (I*4)   IZIN    = ION CHARGE OF EMITTING ION
C  INPUT : (I*4)   IZ0IN   = NUCLEAR CHARGE OF EMITTING ION
C
C  INPUT : (I*4)   ITVAL   = NO. OF ELECTRON TEMPERATURE VALUES
C  INPUT : (I*4)   IDVAL   = NO. OF ELECTRON DENSITY VALUES
C  INPUT : (R*8)   TVAL()  = ELECTRON TEMPERATURES (UNITS: EV)
C                            DIMENSION: TEMPERATURE INDEX
C  INPUT : (R*8)   DVAL()  = ELECTRON DENSITIES (UNITS: CM-3)
C                            DIMENSION: DENSITY INDEX
C
C  OUTPUT: (R*8)   WLNGTH  = SELECTED BLOCK WAVELENGTH (ANGSTROMS)
C
C  OUTPUT: (R*8)   PECA(,) = PHOTON EMISSIVITIES.
C                            1ST DIM: TEMPERATURE INDEX
C                            2ND DIM: DENSITY INDEX
C  OUTPUT: (L*4)   LTRNG() =.TRUE.  => OUTPUT 'PECA()'  VALUE WAS INTER-
C                                      POLATED  FOR  THE  USER  ENTERED
C                                      ELECTRON TEMPERATURE 'TVAL()'.
C                           .FALSE. => OUTPUT 'PECA()'  VALUE WAS EXTRA-
C                                      POLATED  FOR  THE  USER  ENTERED
C                                      ELECTRON TEMPERATURE 'TVAL()'.
C                            DIMENSION: TEMPERATURE INDEX
C  OUTPUT: (L*4)   LDRNG() =.TRUE.  => OUTPUT 'PECA()'  VALUE WAS INTER-
C                                      POLATED  FOR  THE  USER  ENTERED
C                                      ELECTRON DENSITY 'DVAL()'.
C                           .FALSE. => OUTPUT 'PECA()'  VALUE WAS EXTRA-
C                                      POLATED  FOR  THE  USER  ENTERED
C                                      ELECTRON DENSITY 'DVAL()'.
C                            DIMENSION: DENSITY PAIR INDEX
C
C  OUTPUT: (C*120) TITLX   = INFORMATION STRING (DSN ETC.)
C  OUTPUT: (I*4)   IRCODE  = RETURN CODE FROM SUBROUTINE:
C                            0 => NORMAL COMPLETION - NO ERROR DETECTED
C                            1 => DATA SET MEMBER FOR EMITTING ION WITH
```

```
C                                CHARGE 'IZIN' & ION CHARGE 'IZ0IN' CAN
C                                NOT BE FOUND/DOES NOT EXIST.
C                              2 => DISCREPANCY BETWEEN REQUESTED CHARGES
C                                AND THOSE IN INPUT FILE.
C                              3 => THE SELECTED DATA-BLOCK 'IBSEL' IS OUT
C                                OF RANGE OR DOES NOT EXIST.
C                              4 => INVALID VALUE FOR 'IZ0IN' ENTERED.
C                                ('IZ0MIN' <= 'IZ0IN' <= 'IZ0MAX')
C                              5 => INVALID VALUE FOR 'IZIN' ENTERED.
C                                ( 0  <= 'IZIN' <= 99 )
C                              9 => ERROR ENCOUNTERED WHEN TRYING TO OPEN
C                                INPUT DATA-SET.
C
C          (I*4)   NSTORE   = PARAMETER= MAXIMUM NUMBER  OF  DATA-BLOCKS
C                                WHICH CAN BE READ FROM THE INPUT
C                                DATA-SET.
C          (I*4)   NTDIM    = PARAMETER= MAXIMUM NUMBER OF ELECTRON TEMP-
C                                ERATURES THAT CAN BE READ  FROM
C                                AN INPUT DATA-SET DATA-BLOCK.
C          (I*4)   NDDIM    = PARAMETER= MAXIMUM NUMBER OF ELECTRON DENS-
C                                ITIES  THAT  CAN  BE  READ  FROM
C                                AN INPUT DATA-SET DATA-BLOCK.
C          (I*4)   IZ0MIN   = PARAMETER: MIN. ALLOWED VALUE FOR 'IZ0IN'
C          (I*4)   IZ0MAX   = PARAMETER: MAX. ALLOWED VALUE FOR 'IZ0IN'
C
C          (I*4)   IUNIT    = UNIT TO WHICH INPUT DATA SET IS ALLOCATED
C          (I*4)   NBSEL    = TOTAL NUMBER OF DATA-BLOCKS READ FROM INPUT
C                             DATA SET.
C          (I*4)   IZ0      = INPUT FILE - EMITTING ION - NUCLEAR CHARGE
C          (I*4)   IZ       = INPUT FILE - EMITTING ION - CHARGE
C          (I*4)   IZ1      = INPUT FILE - EMITTING ION - CHARGE + 1
C
C          (L*4)   LOPEN    = .TRUE.  => INPUT DATA SET OPEN.
C                             .FALSE. => INPUT DATA SET CLOSED.
C
C          (C*2)   ESYM     = INPUT FILE - EMITTING ION - ELEMENT SYMBOL
C          (C*120) DSNAME   = NAME OF DATA SET INTERROGATED
C
C          (I*4)   ISELA()  = INPUT DATA FILE: DATA-BLOCK ENTRY INDICES.
C                             DIMENSION: DATA-BLOCK INDEX
C          (I*4)   ITA()    = INPUT DATA SET-NUMBER OF ELECTRON
C                             TEMPERATURES.
C                             DIMENSION: DATA-BLOCK INDEX
C          (I*4)   IDA()    = INPUT DATA SET-NUMBER OF ELECTRON DENSITIES
C                             DIMENSION: DATA-BLOCK INDEX
C
C          (R*8)   TETA(,)  = INPUT DATA SET -
C                             ELECTRON TEMPERATURES (UNITS: eV)
C                             1st DIMENSION: ELECTRON TEMPERATURE INDEX
C                             2nd DIMENSION: DATA-BLOCK INDEX
C          (R*8)   TEDA(,)  = INPUT DATA SET -
C                             ELECTRON DENSITIES    (UNITS: cm-3)
C                             1st DIMENSION: ELECTRON DENSITY     INDEX
C                             2nd DIMENSION: DATA-BLOCK INDEX
C          (R*8)   PEC(,,)  = INPUT DATA SET -
C                             FULL SET OF IONIZATIONS PER PHOTON
C                             1st DIMENSION: ELECTRON TEMPERATURE INDEX
C                             2nd DIMENSION: ELECTRON DENSITY     INDEX
C                             3rd DIMENSION: DATA-BLOCK INDEX
C
C          (C*10)  CWAVEL() = INPUT FILE - WAVELENGTH (ANGSTROMS)
```

```
C                               DIMENSION: DATA-BLOCK INDEX
C          (C*8)   CFILE()  = INPUT FILE - SPECIFIC ION FILE SOURCE
C                               DIMENSION: DATA-BLOCK INDEX
C          (C*8)   CTYPE()  = INPUT FILE - TYPE OF DATA (IE EXCIT., ETC)
C                               DIMENSION: DATA-BLOCK INDEX
C          (C*2)   CINDM()  = INPUT FILE - METASTABLE INDEX
C                               DIMENSION: DATA-BLOCK INDEX
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          E3DATA     ADAS      FETCH INPUT DATA FROM SELECTED DATA SET
C          E3CHKB     ADAS      CHECK VALIDITY OF ION AND 'IBSEL'
C          E3SPLN     ADAS      INTERPOLATE DATA WITH TWO WAY SPLINES
C          E3TITL     ADAS      CREATE DESCRIPTIVE TITLE FOR OUTPUT
C
C AUTHOR:  Alessandro Lanzafame, University of Strathclyde.
C
C DATE:    7th December 1995
C
C-----------------------------------------------------------------------
C
C VERSION: 1.1                                          DATE: 12-03-98
C MODIFIED: RICHARD MARTIN
C               - PUT UNDER SCCS CONTROL
C
C VERSION:  1.2                                         DATE: 29-05-2002
C MODIFIED: Martin O'Mullane
C               - Change dimension of arrays from e3data as the 96 pecs
C                 can have 24 entries. It is not necessary to tie
C                 these dimensions to the global NTDIM/NDDIM used
C                 in the rest of the program.
C
C-----------------------------------------------------------------------

C-----------------------------------------------------------------------
      CHARACTER*120       DSNAME,      TITLX
      INTEGER             IBSEL,       IDVAL,       IRCODE,      ITVAL
      INTEGER             IZ0IN,       IZIN
      LOGICAL             LDRNG(IDVAL),             LTRNG(ITVAL)
      REAL*8              DVAL(NDDIM), PECA(NTDIM,NDDIM)
      REAL*8              TVAL(NTDIM), WLNGTH
```

5.65 d9spec: Subroutine d9spec from library adas4xx

```
C Copyright (c) 1997, Strathclyde University.
      SUBROUTINE D9SPEC( LRSPEC ,
     &                   NDLINE , NDCOMP  , NDRAT  , NDFILE ,
     &                   NFILE  , LFILE   ,
     &                   UID    , GROUP   , TYPE   , EXT    ,
     &                   IZ0    , DSPECA  ,
     &                   NLINE  , NCOMP   ,
     &                   IZION  , IMET    , CIMET  , INDPH  ,
     &                   IFILE  ,
     &                   NTDIM  , NDDIM   , ITMAX  , IDMAX  ,
     &                   TEIN   , DEIN    , THIN   , DHIN   ,
     &                   PECA   ,
     &                   LPEC   , LTRNG   , LDRNG
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: D9SPEC  *******************
C
C  PURPOSE: TO CALCULATE PHOTON EMISSIVITY COEFFICIENTS FOR
C           SPECTRAL LINES IDENTIFIED IN SCRIPT FILE
C
C  CALLING PROGRAM: ADAS409
C
C  SUBROUTINE:
C
C  INPUT : (I*4)   NDLINE   = MAXIMUM NUMBER OF LINES ALLOWED
C  INPUT : (I*4)   NDCOMP   = MAXIMUM NUMBER OF COMPONENT FOR EACH LINE
C  INPUT : (I*4)   NDRAT    = MAXIMUM NUMBER OF LINE RATIOS ALLOWED
C  INPUT : (I*4)   NDFILE   = MAXIMUM NUMBER OF EMISSIVITY FILES WHICH
C                             CAN BE SEARCHED
C  INPUT : (I*4)   NFILE    = NUMBER OF PEC FILES TO BE SCANNED
C  INPUT : (L*4)   LFILE()  = .TRUE.  => PEC FILE EXISTS AND MATCHES
C                             .FALSE. => PEC FILE DOES NOT EXIST/MATCH
C  INPUT : (C*6)   UID()    = USER IDENTIFIER OF PEC FILE
C  INPUT : (C*8)   GROUP()  = GROUP IDENTIFIER OF PEC FILE
C  INPUT : (C*5)   TYPE()   = TYPE IDENTIFIER OF PEC FILE
C  INPUT : (C*3)   EXT()    = EXTENSION OF PEC FILE MEMBER NAME
C  INPUT : (I*4)   IZ0      = NUCLEAR CHARGE OF IMPURITY
C  INPUT : (C*120)DSPECA()  = PHOTON EMISSIVITY SOURCE FILES
C  INPUT : (I*4)   NLINE    = NUMBER OF LINES IDENTIFIED IN SCRIPT
C  INPUT : (I*4)   NCOMP()  = NUMBER OF COMPONENTS OF SCRIPT LINE
C                             1ST DIM: LINE INDEX
C  INPUT : (I*4)   IZION(,) = CHARGE STATE OF COMPONENT
C                             1ST DIM: LINE INDEX
C                             2ND DIM: COMPONENT INDEX
C  INPUT : (I*4)   IMET(,)  = NUMBER OF COMPONENTS OF SCRIPT LINE
C                             1ST DIM: LINE INDEX
C                             2ND DIM: COMPONENT INDEX
C  INPUT : (C*1)   CIMET(,) = SIGN (+, BLANK OR -) OF METASTABLE
C                             1ST DIM: LINE INDEX
C                             2ND DIM: COMPONENT INDEX
C  INPUT : (I*4)   INDPH(,) = PEC FILE INDEX OF LINE COMPONENT
C                             1ST DIM: LINE INDEX
C                             2ND DIM: COMPONENT INDEX
C  INPUT : (I*4)   IFILE(,) = INDEX OF PEC FILE IN FILE LIST
C                             1ST DIM: LINE INDEX
C                             2ND DIM: COMPONENT INDEX
C  INPUT : (I*4)   NTDIM    = MAXIMUM NUMBER OF TEMPERATURE SETS
C  INPUT : (I*4)   NDDIM    = MAXIMUM NUMBER OF DENSITY SETS
```

```
C  INPUT : (I*4)   ITMAX    = NUMBER OF TEMPERATURE SETS
C  INPUT : (I*4)   IDMAX    = NUMBER OF DENSITY SETS
C  INPUT : (R*8)   TEIN()   = ELECTRON TEMPERATURES (EV)
C  INPUT : (R*8)   DEIN()   = ELECTRON DENSITIES (CM-3)
C  INPUT : (R*8)   THIN()   = HYDROGEN TEMPERATURES (EV)
C  INPUT : (R*8)   DHIN()   = HYDROGEN DENSITIES (CM-3)
C
C  OUTPUT: (L*4)   LRSPEC   = .TRUE.  => PEC PROCESSING DONE
C                             .FALSE. => PEC PROCESSING NOT DONE
C  OUTPUT: (R*8)   PECA(,,,)= PHOTON EMISSIVITY COEFFICIENTS (CM3 S-1)
C                             1ST DIM: TEMPERATURE INDEX
C                             2ND DIM  DENSITY INDEX
C                             3RD DIM: LINE INDEX
C                             4RD DIM: COMPONENT INDEX
C  OUTPUT: (L*4)   LPEC(,)  = .TRUE.  => PHOTON EMISSIVITY OBTAINED
C                             .FALSE. => PHOTON EMISSIVITY NOT OBTAINED
C                             2ND DIM: LINE INDEX
C                             3RD DIM: COMPONENT INDEX
C
C          (I*4)   IUNT10  = PARAMETER = INPUT UNIT FOR DATA
C          (L*4)   OPEN10  = .TRUE.  => FILE ALLOCATED TO UNIT 10.
C                            .FALSE. => NO FILE ALLOCATED TO UNIT 10.
C
C ROUTINES:
C          ROUTINE   SOURCE   BRIEF DESCRIPTION
C          ------------------------------------------------------------
C    D9SPC2   IDL-ADAS  OBTAIN PHOTON EMISSIVITY COEFFICIENT
C
C AUTHOR:  A. C. Lanzafame, University of Strathclyde
C
C DATE:    7th December 1995
C
C-----------------------------------------------------------------------
C
C VERSION: 1.1                                          DATE: 12-03-98
C MODIFIED: RICHARD MARTIN
C               - PUT UNDER SCCS CONTROL
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      CHARACTER           CIMET(NDLINE,NDCOMP)
      CHARACTER*120       DSPECA(NDFILE)
      CHARACTER*3         EXT(NDFILE)
      CHARACTER*8         GROUP(NDFILE)
      CHARACTER*5         TYPE(NDFILE)
      CHARACTER*6         UID(NDFILE)
      INTEGER             IDMAX,       IFILE(NDLINE,NDCOMP)
      INTEGER             IMET(NDLINE,NDCOMP),      INDPH(NDLINE,NDCOMP)
      INTEGER             ITMAX,       IZ0,         IZION(NDLINE,NDCOMP)
      INTEGER             NCOMP(NDLINE),            NDCOMP,      NDDIM
      INTEGER             NDFILE,      NDLINE,      NDRAT,       NFILE
      INTEGER             NLINE,       NTDIM
      LOGICAL             LDRNG(NTDIM),             LFILE(NDFILE)
      LOGICAL             LPEC(NDLINE,NDCOMP),      LRSPEC
      LOGICAL             LTRNG(NTDIM)
      REAL*8              DEIN(NDDIM), DHIN(NDDIM)
      REAL*8              PECA(NTDIM,NDDIM,NDLINE,NDCOMP)
      REAL*8              TEIN(NTDIM), THIN(NTDIM)
```

5.66 d9spln: Subroutine d9spln from library adas4xx

```
C Copyright (c) 1997, Strathclyde University.
      SUBROUTINE D9SPLN( NTDIM  , NDDIM  ,
     &                   ITA    , IDA    , ITVAL   , IDVAL  ,
     &                   TETA   , TEDA   , TEVA    , DIN    ,
     &                   PEC    ,          PECA    ,
     &                                     LTRNG   , LDRNG
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: D9SPLN *********************
C
C  PURPOSE:
C          PERFORMS CUBIC SPLINE OF LOG(PHOTON EMISSIVITY COEFFICIENTS)
C          ON 2D GRID (LOG(TEMPERATURE), LOG(DENSITY))
C          INPUT DATA FOR A GIVEN WAVELENGTH DATA-BLOCK.
C
C          USING  TWO-WAY SPLINES IT CALCULATES  THE  PHOTON EMISSIVITY
C          FOR  'ITVAL' AND 'IDVAL'  INDEX OF  ELECTRON TEMPERATURES
C          AND  DENSITIES RESPECTIVELY.
C          FROM THE TWO-DIMENSIONAL TABLE OF TEMPERATURES/DENSITIES READ
C          IN FROM THE INPUT FILE. IF A  VALUE  CANNOT  BE  INTERPOLATED
C          USING SPLINES IT IS EXTRAPOLATED VIA 'XXSPLE'.
C
C  CALLING PROGRAM: ADAS409
C
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  NTDIM   = MAX NUMBER OF ELECTRON TEMPERATURES ALLOWED
C  INPUT : (I*4)  NDDIM   = MAX NUMBER OF ELECTRON DENSITIES    ALLOWED
C
C  INPUT : (I*4)  ITA     = INPUT DATA FILE: NUMBER OF ELECTRON TEMPERA-
C                           TURES READ FOR THE DATA-BLOCK BEING ASSESSED
C  INPUT : (I*4)  IDA     = INPUT DATA FILE: NUMBER OF ELECTRON DENSIT-
C                           IES   READ FOR THE DATA-BLOCK BEING ASSESSED
C  INPUT : (I*4)  ITVAL   = NUMBER OF ISPF ENTERED TEMPERATURE/DENSITY
C                           PAIRS  FOR  WHICH  IOINIZATIONS PER PHOTON
C                           ARE REQUIRED FOR TABULAR/GRAPHICAL OUTPUT.
C
C  INPUT : (R*8)  TETA()  = INPUT DATA FILE: ELECTRON TEMPERATURES (EV)
C                           FOR THE DATA-BLOCK BEING ASSESSED.
C                           DIMENSION: ELECTRON TEMPERATURE INDEX
C  INPUT : (R*8)  TEDA()  = INPUT DATA FILE: ELECTRON DENSITIES (CM-3)
C                           FOR THE DATA-BLOCK BEING ASSESSED.
C                           DIMENSION: ELECTRON DENSITY INDEX
C  INPUT : (R*8)  TEVA()  = USER ENTERED: ELECTRON TEMPERATURES (EV)
C                           DIMENSION: TEMPERATURE INDEX
C  INPUT : (R*8)  DIN()   = USER ENTERED: ELECTRON DENSITIES (CM-3)
C                           DIMENSION: DENSITY INDEX
C
C  INPUT : (R*8)  PEC(,)   =INPUT DATA FILE: FULL SET OF IONIZATIONS PER
C                           PHOTON VALUES FOR THE DATA-BLOCK BEING
C                           ANALYSED.
C                           1ST DIMENSION: ELECTRON TEMPERATURE INDEX
C                           2ND DIMENSION: ELECTRON DENSITY     INDEX
C  OUTPUT: (R*8)  PECA(,) = SPLINE INTERPOLATED OR  EXTRAPOLATED PHOTON
C                           EMISSIVITY COEFFICIENT AT THE USER ENTERED
C                           ELECTRON TEMPERATURE AND DENSITY POINTS.
C                           1ST DIM: TEMPERATURE INDEX
```

```
C                          2ND DIM: DENSITY INDEX
C
C  OUTPUT: (L*4)  LTRNG()=  .TRUE.  => OUTPUT 'PECA(,)' VALUE WAS INTER-
C                                       POLATED  FOR  THE  USER  ENTERED
C                                       ELECTRON TEMPERATURE 'TEVA()'.
C                          .FALSE. => OUTPUT 'PECA()' VALUE WAS EXTRA-
C                                       POLATED  FOR  THE  USER  ENTERED
C                                       ELECTRON TEMPERATURE 'TEVA()'.
C                          DIMENSION: TEMPERATURE INDEX
C
C  OUTPUT: (L*4)  LDRNG()=  .TRUE.  => OUTPUT 'PECA(,)' VALUE WAS INTER-
C                                       POLATED  FOR  THE  USER  ENTERED
C                                       ELECTRON DENSITY 'DIN()'.
C                          .FALSE. => OUTPUT 'PECA()' VALUE WAS EXTRA-
C                                       POLATED  FOR  THE  USER  ENTERED
C                                       ELECTRON DENSITY 'DIN()'.
C                          DIMENSION: DENSITY INDEX
C
C          (I*4)  NTIN    = PARAMETER = MAX. NO. OF INPUT  TEMPERATURE
C                                       VALUES. MUST BE >= 'ITA'
C          (I*4)  NDIN    = PARAMETER = MAX. NO. OF INPUT  DENSITY
C                                       VALUES. MUST BE >= 'IDA'
C          (I*4)  NTOUT   = PARAMETER = MAX. NO. OF OUTPUT TEMPERATURE
C                                       VALUES.  MUST BE >= 'ITVAL'
C          (I*4)  NDOUT   = PARAMETER = MAX. NO. OF OUTPUT DENSITY
C                                       PAIRS.  MUST BE >= 'IDVAL'
C          (I*4)  L1      = PARAMETER = 1
C
C          (I*4)  IED     = ARRAY SUBSCRIPT USED INPUT  FILE  ELECTRON
C                           DENSITIES.
C          (I*4)  IET     = ARRAY SUBSCRIPT USED INPUT  FILE  ELECTRON
C                           TEMPERATURES.
C          (I*4)  IT      = ARRAY  SUBSCRIPT  USED  FOR  USER  ENTERED
C                           TEMPERATURE INDEX .
C          (I*4)  ID      = ARRAY  SUBSCRIPT  USED  FOR  USER  ENTERED
C                           DENSITY INDEX .
C          (I*4)  IOPT    = DEFINES THE BOUNDARY DERIVATIVES FOR THE
C                           SPLINE ROUTINE 'XXSPLE', SEE 'XXSPLE'.
C                           (VALID VALUES = <0, 0, 1, 2, 3, 4)
C
C          (L*4)  LSETX   = .TRUE.  => SET UP SPLINE PARAMETERS RELATING
C                                      TO 'XIN' AXIS.
C                           .FALSE. => DO NOT SET UP SPLINE PARAMETERS
C                                      RELATING TO 'XIN' AXIS.
C                                      (I.E. THEY WERE SET IN A PREVIOUS
C                                            CALL )
C                           (VALUE SET TO .FALSE. BY 'XXSPLE')
C
C          (R*8)  R8FUN1  = FUNCTION - (SEE ROUTINES SECTION BELOW)
C
C          (R*8)  XIN()   = 1) LOG( DATA FILE ELECTRON DENSITIES    )
C                           2) LOG( DATA FILE ELECTRON TEMPERATURES )
C          (R*8)  YIN()   = LOG( DATA FILE IONIZATIONS/PHOTON )
C          (R*8)  XOUT()  = 1) LOG( SCALED USER ENTERED ELECTRON DENS. )
C                           2) LOG( SCALED USER ENTERED ELECTRON TEMPS.)
C          (R*8)  YOUT()  = LOG( OUTPUT GENERATED IONIZATIONS/PHOTON )
C          (R*8)  YPASS(,)= LOG( IONIZATIONS/PHOTON) INTERMEDIATE ARRAY
C                           WHICH   STORES   INTERPOLATED/EXTRAPOLATED
C                           VALUES  BETWEEN  THE  TWO  SPLINE SECTIONS.
C                           SECTIONS.
C          (R*8)  DFT()   = SPLINE INTERPOLATED DERIVATIVES (TEMPERATURE)
```

```
C          (R*8)  DFD()   = SPLINE INTERPOLATED DERIVATIVES (DENSITY)
C
C
C NOTE:
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          XXSPLE     ADAS      SPLINE SUBROUTINE (EXTENDED DIAGNOSTICS)
C          R8FUN1     ADAS      REAL*8 FUNCTION: ( X -> X )
C
C AUTHOR:  Alessandro Lanzafame, University od Strathclyde
C
C DATE:    7th December 1995
C
C-----------------------------------------------------------------------
C
C VERSION: 1.1                                          DATE: 12-03-98
C MODIFIED: RICHARD MARTIN
C               - PUT UNDER SCCS CONTROL
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IDA,         IDVAL,       ITA,         ITVAL
      INTEGER             NDDIM,       NTDIM
      LOGICAL             LDRNG(IDVAL),             LTRNG(ITVAL)
      REAL*8              DIN(NTDIM),  PEC(NTDIM,NDDIM)
      REAL*8              PECA(NTDIM,NTDIM),        TEDA(IDA)
      REAL*8              TETA(ITA),   TEVA(NTDIM)
```

5.67 d9spow: Subroutine d9spow from library adas4xx

```
C Copyright (c) 1997, Strathclyde University.
      SUBROUTINE D9SPOW( LSELA  , LEXSA  , LDEFA  , LPART  , LEXSS  ,
     &                   IZ0    , IZL    , IZH    , NPART  ,
     &                   ISDIMD , IZDIMD , ITDIMD , IPDIMD , IMDIMD ,
     &                   ACDA   , SCDA   , CCDA   , PRBA   ,
     &                   PRCA   , QCDA   , XCDA   , PLTA   ,
     &                   NMSUM  , IZIP   , IMIP   , IPIZM  ,
     &                   NTDIM  , NDDIM  , ITMAX  , IDMAX  ,
     &                   DENS   , DENSH  ,
     &                   FPABUN , FSABUN ,
     &                   PLTPEQ ,
     &                   ACDSEQ , SCDSEQ , CCDSEQ , PRBSEQ ,
     &                   PRCSEQ , PLTSEQ ,
     &                   PRBEQ  , PRCEQ  , PLTEQ  , PRADA
     &                  )
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: D9SPOW *********************
C
C PURPOSE : TO ASSEMBLE RADIATED POWER FUNCTIONS USING FRACTIONAL
C           METASTABLE ABUNDANCES.
C           GENERATE STANDARD ISONUCLEAR MASTER DATA FROM PARTIAL DATA.
C           2D (TEMPERATURE, DENSITY) VERSION.
C
C NOTE    : THE SOURCE ISONUCLEAR MASTER FILE DATA ARE OBTAINED BY A
C           PRIOR CALL TO SUBROUTINE D9DATA FROM SEQUENTIAL FILES
C           WITH THE FOLLOWING NAMING CONVENTIONS:
C
C                   (1) JETSHP.ACD<YR>#<EL).<CODE>DATA
C                   (2) JETSHP.SCD<YR>#<EL>.<CODE>DATA
C                   (3) JETSHP.CCD<YR>#<EL>.<CODE>DATA
C                   (4) JETSHP.PRB<YR>#<EL>.<FILT>.<CODE>DATA
C                   (5) JETSHP.PRC<YR>#<EL>.<FILT>.<CODE>DATA
C                   (6) JETSHP.QCD<YR>#<EL>.<CODE>DATA
C                   (7) JETSHP.XCD<YR>#<EL>.<CODE>DATA
C                   (8) JETSHP.PLT<YR>#<EL>.<CODE>DATA
C
C       WHERE, <YR>   = TWO DIGIT YEAR NUMBER
C              <EL>   = ONE OR TWO CHARACTER ELEMENT SYMBOL
C              <CODE> = R       => PARTIAL DATA
C                       U       => PARTIAL DATA
C                       OMITTED => STANDARD DATA
C              <FILT> = SIX CHARACTER POWER FILTER CODE
C
C       AND DATA OF CLASSES 6 AND 7 DO NOT EXIST FOR THE PARTIAL CASE.
C
C
C INPUT  : (L*4)  LSELA()   = .TRUE.  => INPUT DATA SET TYPE FOR THIS
C                                        INDEX SELECTED
C                           = .FALSE. => INPUT DATA SET FOR THIS INDEX
C                                        NOT SELECTED
C INPUT  : (L*4)  LEXSA()   = .TRUE.  => INPUT DATA SET TYPE FOR THIS
C                                        SELECTED INDEX EXISTS
C                           = .FALSE. => INPUT DATA SET DOES NOT EXIST
C                                        FOR THIS SELECTED INDEX
C INPUT  : (L*4)  LDEFA()   = .TRUE.  => INPUT DATA SET TYPE FOR THIS
C                                        DEFAULT YEAR INDEX EXISTS
C                           = .FALSE. => INPUT DATA SET DOES NOT EXIST
```

```
C                                          FOR THIS DEFAULT YEAR INDEX
C INPUT  : (L*4)  LPART     = .TRUE.  => PARTIAL DATA SELECTED
C                           = .FALSE. => STANDARD DATA SELECTED
C INPUT  : (I*4)  IZ0       = NUCLEAR CHARGE
C INPUT  : (I*4)  IZL       = MINIMUM ION CHARGE+1 IN MASTER DATA FILES
C INPUT  : (I*4)  IZH       = MAXIMUM ION CHARGE+1 IN MASTER DATA FILES
C INPUT  : (I*4)  NPART()   = METASTABLE PARTITION.  I.E. NUMBER OF
C                             METASTABLES FROM CHARGE STATE IZL-1 TO
C                             IZH ON INPUT
C INPUT  : (I*4)  ISDIMD    = MAXIMUM NUMBER OF (CHARGE, PARENT, GROUND)
C                             BLOCKS IN ISONUCLEAR MASTER FILES
C INPUT  : (I*4)  IZDIMD    = MAXIMUM NUMBER OF CHARGE STATES
C                             IN ISONUCLEAR MASTER FILES
C INPUT  : (I*4)  ITDIMD    = MAXIMUM NUMBER OF TEMP OR DENS VALUES IN
C                             ISOELECTRONIC MASTER FILES
C INPUT  : (I*4)  IPDIMD    = MAXIMUM NUMBER OF METASTABLES FOR EACH
C                             IONISATION STAGE
C INPUT  : (I*4)  IMDIMD    = MAXIMUM NUMBER OF METASTABLES
C
C INPUT  : (R*8)  ACDA(,,,,)= INTERPOLATION OF ACD COEFFICIENT (CM3 S-1)
C                             1ST DIM: TEMPERATURE INDEX
C                             2ND DIM: DENSITY INDEX
C                             3RD DIM: CHARGE STATE INDEX
C                             4TH DIM: RECOMBINING METASTABLE INDEX
C                             5TH DIM: RECOMBINED METASTABLE INDEX
C INPUT  : (R*8)  SCDA(,,,,)= INTERPOLATION OF SCD COEFFICIENT (CM3 S-1)
C                             1ST DIM: TEMPERATURE INDEX
C                             2ND DIM: DENSITY INDEX
C                             3RD DIM: CHARGE STATE INDEX
C                             4TH DIM: RECOMBINING METASTABLE INDEX
C                             5TH DIM: RECOMBINED METASTABLE INDEX
C INPUT  : (R*8)  CCDA(,,,,)= INTERPOLATION OF CCD COEFFICIENT (CM3 S-1)
C                             1ST DIM: TEMPERATURE INDEX
C                             2ND DIM: DENSITY INDEX
C                             3RD DIM: CHARGE STATE INDEX
C                             4TH DIM: RECOMBINING METASTABLE INDEX
C                             5TH DIM: RECOMBINED METASTABLE INDEX
C INPUT  : (R*8)  PRBA(,,,) = INTERPOLATION OF PRB COEFFICIENT (W CM3 )
C                             1ST DIM: TEMPERATURE INDEX
C                             2ND DIM: DENSITY INDEX
C                             3RD DIM: CHARGE STATE INDEX
C                             4TH DIM: RECOMBINING METASTABLE INDEX
C INPUT  : (R*8)  PRCA(,,,) = INTERPOLATION OF PRC COEFFICIENT (W CM3 )
C                             1ST DIM: TEMPERATURE INDEX
C                             2ND DIM: DENSITY INDEX
C                             3RD DIM: CHARGE STATE INDEX
C                             4TH DIM: RECOMBINING METASTABLE INDEX
C INPUTT : (R*8)  QCDA(,,,) = INTERPOLATION OF QCD COEFFICIENT (CM3 S-1)
C                             1ST DIM: TEMPERATURE INDEX
C                             2ND DIM: DENSITY INDEX
C                             3RD DIM: CHARGE STATE INDEX
C                             4TH DIM: FIRST METASTABLE INDEX
C                             5TH DIM: SECOND METASTABLE INDEX
C INPUT  : (R*8)  XCDA(,,,) = INTERPOLATION OF XCD COEFFICIENT (CM3 S-1)
C                             1ST DIM: TEMPERATURE INDEX
C                             2ND DIM: DENSITY INDEX
C                             3RD DIM: CHARGE STATE INDEX
C                             4TH DIM: FIRST PARENT METASTABLE INDEX
C                             5TH DIM: SECOND PARENT METASTABLE INDEX
C INPUT  : (R*8)  PLTA(,,,) = INTERPOLATION OF PLT COEFFICIENT (W CM3 )
C                             1ST DIM: TEMPERATURE INDEX
```

```
C                               2ND DIM: DENSITY INDEX
C                               3RD DIM: CHARGE STATE INDEX
C                               4TH DIM: METASTABLE INDEX
C INPUT  : (I*4)  NMSUM     = TOTAL NUMBER OF POPULATIONS
C
C INPUT  :        IZIP()    = ION CHARGE +1 (IZ1) OF METASTABLE IN LIST
C INPUT  :        IMIP()    = METASTABLE INDEX WITHIN CHARGE STATE IZ1
C                               OF METASTABLE INDEX FROM COMPLETE LIST
C INPUT  :        IPIZM(,)  = METASTABLE INDEX IN COMPLETE LIST
C                               1ST DIM: INDEX IZ1-IZL+1
C                               2ND DIM: METASTABLE COUNT FOR STAGE (IGRD)
C INPUT  : (I*4)  NTDIM     = MAXIMUM NUMBER OF DTEV/DDENS PAIRS
C INPUT  : (I*4)  ITMAX     = NUMBER OF ( DTEV() , DDENS() ) PAIRS
C INPUT  : (R*8)  DENS()    = ELECTRON DENSITIES (CM-3))
C INPUT  : (R*8)  DENSH()   = HYDROGEN DENSITIES (CM-3))
C INPUT  : (R*8)  FPABUN(,,)= RESOLVED METASTABLE EQUILIBRIUM
C                               FRACTIONAL ABUNDANCES
C                               1ST DIM: - TEMPERATURE INDEX
C                               2ND DIM: - DENSITY INDEX
C                               3RD DIM: - METASTABLE INDEX
C OUTPUT : (L*4)  LEXSS()   = .TRUE.  => OUTPUT STANDARD MASTER DATA FOR
C                                        THIS INDEX GENERATED
C                             .FALSE. => OUTPUT STANDARD MASTER DATA FOR
C                                        THIS INDEX NOT GENERATED
C OUTPUT : (R*8)  FSABUN(,,)= STAGE EQUILIBRIUM FRACTIONAL ABUNDANCES
C                               1ST DIM: - TEMPERATURE INDEX
C                               2ND DIM: - DENSITY INDEX
C                               3RD DIM: - CHARGE STATE INDEX (IZ1-IZL+1)
C OUTPUT : (R*8)  PLTPEQ(,,)= METASTABLE PARTIAL EQUILIBRIUM RADIATED
C                               LINE POWER FUNCTIONS
C                               1ST DIM: - TEMPERATURE INDEX
C                               2ND DIM: - DENSITY INDEX
C                               3RD DIM: - METASTABLE INDEX
C OUTPUT : (R*8)  ACDSEQ(,,)= STANDARD (UNRESOLVED) ACD COEFFICIENT
C                               1ST DIM: - TEMPERATURE INDEX
C                               2ND DIM: - DENSITY  INDEX
C                               3RD DIM: - CHARGE STATE INDEX (IZ1-IZL+1)
C OUTPUT : (R*8)  SCDSEQ(,,)= STANDARD (UNRESOLVED) SCD COEFFICIENT
C                               1ST DIM: - TEMPERATURE INDEX
C                               2ND DIM: - DENSITY INDEX
C                               3RD DIM: - CHARGE STATE INDEX (IZ1-IZL+1)
C OUTPUT : (R*8)  CCDSEQ(,,)= STANDARD (UNRESOLVED) CCD COEFFICIENT
C                               1ST DIM: - TEMPERATURE INDEX
C                               2ND DIM: - DENSITY INDEX
C                               3RD DIM: - CHARGE STATE INDEX (IZ1-IZL+1)
C OUTPUT : (R*8)  PRBSEQ(,,)= STANDARD (UNRESOLVED) SCD COEFFICIENT
C                               1ST DIM: - TEMPERATURE INDEX
C                               2ND DIM: - DENSITY INDEX
C                               3RD DIM: - CHARGE STATE INDEX (IZ1-IZL+1)
C OUTPUT : (R*8)  PRCSEQ(,,)= STANDARD (UNRESOLVED) CCD COEFFICIENT
C                               1ST DIM: - TEMPERATURE INDEX
C                               2ND DIM: - DENSITY INDEX
C                               3RD DIM: - CHARGE STATE INDEX (IZ1-IZL+1)
C OUTPUT : (R*8)  PLTSEQ(,,)= STANDARD (UNRESOLVED) CCD COEFFICIENT
C                               1ST DIM: - TEMPERATURE INDEX
C                               2ND DIM: - DENSITY INDEX
C                               3RD DIM: - CHARGE STATE INDEX (IZ1-IZL+1)
C OUTPUT : (R*8)  PRBEQ(,)  = TOTAL EQUILIBRIUM RADIATED RECOM-BREMS
C                               POWER FUNCTION
C                               1ST DIM: - TEMPERATURE INDEX
C                               2ND DIM: - DENSITY INDEX
```

```
C OUTPUT : (R*8)  PRCEQ(,)  = TOTAL EQUILIBRIUM CX RADIATED RECOM POWER
C                             FUNCTION NORMALISED TO ELECTRON
C                             DENSITY
C                             1ST DIM: - TEMPERATURE INDEX
C                             2ND DIM: - DENSITY INDEX
C OUTPUT : (R*8)  PLTEQ(,)  = TOTAL EQUILIBRIUM RADIATED LINE POWER
C                             FUNCTION
C                             1ST DIM: - TEMPERATURE INDEX
C                             2ND DIM: - DENSITY INDEX
C OUTPUT : (R*8)  PRADA(,)  = TOTAL EQUILIBRIUM RADIATED POWER FUNCTION
C                             1ST DIM: - TEMPERATURE INDEX
C                             2ND DIM: - DENSITY INDEX
C
C PROGRAM: (I*4)  IT        = GENERAL INDEX FOR TEMPERATURE
C          (I*4)  ID        = GENERAL INDEX FOR DENSITY
C          (I*4)  IZ        = GENERAL INDEX FOR CHARGE
C          (I*4)  IP        = GENERAL INDEX FOR CHARGE
C          (I*4)  IZ1       = GENERAL INDEX FOR CHARGE+1
C          (I*4)  ICL       = GENERAL INDEX FOR CLASS
C          (I*4)  IPP       = GENERAL PARENT INDEX
C          (I*4)  IPG       = GENERAL GROUND INDEX
C          (I*4)  IZREF     = GENERAL CHARGE STAE POINTER INDEX
C          (I*4)  IPRT      = GENERAL INDEX FOR PARENT METASTABLE
C          (I*4)  IGRD      = GENERAL INDEX FOR METASTABLE
C
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C
C
C AUTHOR : Alessandro Lanzafame, University of Strathclyde
C
C DATE   : 11 December 1995
C
C-----------------------------------------------------------------------
C
C VERSION: 1.1                                          DATE: 12-03-98
C MODIFIED: RICHARD MARTIN
C               - PUT UNDER SCCS CONTROL
C
C-----------------------------------------------------------------------
      INTEGER             IDMAX,       IMDIMD,      IMIP(IMDIMD)
      INTEGER             IPDIMD,      IPIZM(IZDIMD,IPDIMD),     ISDIMD
      INTEGER             ITDIMD,      ITMAX,       IZ0,         IZDIMD
      INTEGER             IZH,         IZIP(IMDIMD),             IZL
      INTEGER             NDDIM,       NMSUM,       NPART(IZDIMD)
      INTEGER             NTDIM
      LOGICAL             LDEFA(8),    LEXSA(8),    LEXSS(8),    LPART
      LOGICAL             LSELA(8)
      REAL*8              ACDA(NTDIM,NDDIM,IZDIMD,IPDIMD,IPDIMD)
      REAL*8              ACDSEQ(NTDIM,NDDIM,IZDIMD)
      REAL*8              CCDA(NTDIM,NDDIM,IZDIMD,IPDIMD,IPDIMD)
      REAL*8              CCDSEQ(NTDIM,NDDIM,IZDIMD)
      REAL*8              DENS(NDDIM), DENSH(NDDIM)
      REAL*8              FPABUN(NTDIM,NDDIM,IMDIMD)
      REAL*8              FSABUN(NTDIM,NDDIM,IZDIMD)
      REAL*8              PLTA(NTDIM,NDDIM,IZDIMD,IPDIMD)
      REAL*8              PLTEQ(NTDIM,NDDIM)
      REAL*8              PLTPEQ(NTDIM,NDDIM,IMDIMD)
      REAL*8              PLTSEQ(NTDIM,NDDIM,IZDIMD)
```

```
REAL*8              PRADA(NTDIM,NDDIM)
REAL*8              PRBA(NTDIM,NDDIM,IZDIMD,IPDIMD)
REAL*8              PRBEQ(NTDIM,NDDIM)
REAL*8              PRBSEQ(NTDIM,NDDIM,IZDIMD)
REAL*8              PRCA(NTDIM,NDDIM,IZDIMD,IPDIMD)
REAL*8              PRCEQ(NTDIM,NDDIM)
REAL*8              PRCSEQ(NTDIM,NDDIM,IZDIMD)
REAL*8              QCDA(NTDIM,NDDIM,IZDIMD,IPDIMD,IPDIMD)
REAL*8              SCDA(NTDIM,NDDIM,IZDIMD,IPDIMD,IPDIMD)
REAL*8              SCDSEQ(NTDIM,NDDIM,IZDIMD)
REAL*8              XCDA(NTDIM,NDDIM,IZDIMD,IPDIMD,IPDIMD)
```

5.68 d9wr11: Subroutine d9wr11 from library adas4xx

```
C Copyright (c) 1997, Strathclyde University.
      SUBROUTINE D9WR11( IUNIT  , DSNINC , DSNMTR , ELEMT ,
     &                   UID    , DATE   , IZ0    ,
     &                   NDLINE , NLINE  ,
     &                   TITL   , IZION  , CIMET  ,
     &                   NTDIM  , NDDIM  , ITMAX  , IDMAX ,
     &                   TEV    , DENS   ,
     &                   GCF
     &                 )
C-----------------------------------------------------------------------
C
C  ******************** FORTRAN77 SUBROUTINE: D9WR11 ******************
C
C  PURPOSE:  TO OUTPUT DATA TO GCF PASSING FILE.
C
C  CALLING PROGRAM: ADAS409
C
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  IUNIT   = OUTPUT UNIT NUMBER FOR RESULTS
C  INPUT : (C*80) DSNINC  = INPUT SCRIPT DATA SET NAME
C  INPUT : (C*80) DSNMTR  = INPUT ACD MASTER FILE NAME
C  INPUT : (C*2)  ELEMT   = ELEMENT SYMBOL.
C
C  INPUT : (C*6)  UID     = USER IDENIFIER
C  INPUT : (C*8)  DATE    = CURRENT DATE
C  INPUT : (I*4)  IZ0     = NUCLEAR CHARGE
C
C  INPUT : (I*4)  NDLINE  = MAXIMUM NUMBER OF SPECTRUM LINES
C  INPUT : (I*4)  NLINE   = NUMBER OF SPECTRUM LINES
C  INPUT : (C*12) TITL()  = TITLE FOR 1ST COMPONENT OF LINE
C  INPUT : (I*4)  IZION() = ION CHARGE FOR 1ST COMPONENT OF LINE
C  INPUT : (C*1)  CIMET() = +/- SHIFT OF ION CHARGE - 1ST COMPONENT
C
C  INPUT : (I*4)  NTDIM   = MAMXIMUM NUMBER OF INPUT TEMPERATURES
C  INPUT : (I*4)  NDDIM   = MAMXIMUM NUMBER OF INPUT DENSITIES
C  INPUT : (I*4)  ITMAX   = NUMBER OF INPUT TEMPERATURES
C  INPUT : (I*4)  IDMAX   = NUMBER OF INPUT DENSITIES
C  INPUT : (R*8)  TEV()   = ELECTRON TEMPERATURES (UNITS: EV)
C  INPUT : (R*8)  DENS()  = ELECTRON DENSITIES  (UNITS: CM-3)
C
C  INPUT : (R*4)  GCF(,,) = G(TE) FUNCTION (CM3 S-1)
C                           1ST DIM: TEMPERATURE INDEX
C                           2ND DIM: DENSITY INDEX
C                           3RD DIM: LINE INDEX
C
C          (I*4) I         = GENERAL USE
C          (I*4) J         = GENERAL USE
C          (I*4) K         = GENERAL USE
C          (I*4) L         = GENERAL USE
C          (I*4) I1        = GENERAL USE
C          (I*4) I2        = GENERAL USE
C          (I*4) IT        = GENERAL USE
C          (C*80)CLINE     = GENERAL USE
C
C
C ROUTINES: NONE
C
```

```
C AUTHOR:  Alessandro Lanzafame
C
C DATE:    13th December 1995
C
C-----------------------------------------------------------------------
C
C VERSION: 1.1                                          DATE: 12-03-98
C MODIFIED: RICHARD MARTIN
C               - PUT UNDER SCCS CONTROL
C
C VERSION: 1.2                              DATE: 20-11-98
C MODIFIED: RICHARD MARTIN & MARTIN O'MULLANE
C                   - REMOVED SEARCH FOR BRACKETS IN DSNINC AND INITIAL STRING
C                       AS STRING = '        '
C
C-----------------------------------------------------------------------
      CHARACTER           CIMET(NLINE)
      CHARACTER*8         DATE
      CHARACTER*80        DSNINC,      DSNMTR
      CHARACTER*2         ELEMT
      CHARACTER*12        TITL(NLINE)
      CHARACTER*80        UID
      INTEGER             IDMAX,       ITMAX,       IUNIT,       IZ0
      INTEGER             IZION(NLINE),             NDDIM,       NDLINE
      INTEGER             NLINE,       NTDIM
      REAL*8              DENS(NDDIM), GCF(NTDIM,NDDIM,NDLINE)
      REAL*8              TEV(NTDIM)
```

5.69 dadata: Subroutine dadata from library adas4xx

```
      SUBROUTINE  DADATA( IUNIT  , NDPRT  , NDREP , NDLEV ,
     &                    NDAUG  , NDT   ,
     &                    SEQSYM , IZ     , IZ0   , IZ1   ,
     &                    NPRNT  , NPRNTI , NPRNTF, BWNP  ,
     &                    IPA    , CSTRPA , ISPA  , ILPA  , XJPA  ,
     &                    WPA    ,
     &                    IL     , BWNR   ,
     &                    IA     , CSTRGA , ISA   , ILA   , XJA   ,
     &                    WA     ,
     &                    NREP   , IAPRS  , CAPRS , IPAUG ,
     &                    IREPA  , NREPA  , AUGA  , LAUGA ,
     &                    IPRTI  , TPRTI  , ISPRTI, DIELR , LDIELR,
     &                    IPRTF  , TPRTF  , ISPRTF,
     &                    NSYSF  , ISYS   , ISPSYS, DIELN , LDIELN,
     &                    DIELT  ,
     &                    NTE    , TEA
     &                  )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: DADATA *********************
C
C  PURPOSE:  TO FETCH DATA FROM INPUT ADF09 DATA SET.
C
C  CALLING PROGRAM: ADAS204/ADAS212/ADAS410
C
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  IUNIT   = UNIT TO WHICH INPUT FILE IS ALLOCATED
C
C  INPUT : (I*4)  NDPRT   = MAXIMUM NUMBER OF PARENT STATES
C  INPUT : (I*4)  NDREP   = MAX. NUMBER OF REPRESENTATIVE N-SHELLS
C  INPUT : (I*4)  NDLEV   = MAXIMUM NUMBER OF RESOLVED LEVELS
C  INPUT : (I*4)  NDAUG   = MAXIMUM NUMBER OF AUGER RATE INITIAL AND
C                           FINAL PARENT PAIRS
C  INPUT : (I*4)  NDT     = MAX. NUMBER OF ELECTRON TEMPERATURES
C
C  OUTPUT: (C*2)  SEQSYM  = RECOMBINED ION SEQ
C  OUTPUT: (I*4)  IZ      = RECOMBINED ION CHARGE
C  OUTPUT: (I*4)  IZ0     = NUCLEAR CHARGE
C  OUTPUT: (I*4)  IZ1     = RECOMBINING ION CHARGE
C  OUTPUT: (I*4)  NPRNT   = TOTAL NUMBER OF PARENTS
C  OUTPUT: (I*4)  NPRNTI  = NUMBER OF PARENTS WHICH ARE INITIAL PARENTS
C  OUTPUT: (I*4)  NPRNTF  = NUMBER OF PARENTS WHICH ARE FINAL PARENTS
C  OUTPUT: (R*8)  BWNP    = BINDING WAVE NO. OF GROUND PARENT (CM-1)
C  OUTPUT: (I*4)  IPA()   = NUMBER OF PARENT ENERGY LEVELS
C  OUTPUT: (C*18) CSTRPA()= NOMENCL./CONFIG. FOR PARENT LEVEL 'IPA()'
C  OUTPUT: (I*4)  ISPA()  = MULTIPLICITY FOR PARENT LEVEL 'IPA()'
C                           NOTE: (ISPA-1)/2 = QUANTUM NUMBER (SP)
C  OUTPUT: (I*4)  ILPA()  = QUANTUM NUMBER (LP) FOR PARENT LEVEL 'IPA()'
C  OUTPUT: (R*8)  XJPA()  = QUANTUM NUMBER (JP) FOR PARENT LEVEL 'IPA()'
C                           NOTE: (2*XJPA)+1 = STATISTICAL WEIGHT
C  OUTPUT: (R*8)  WPA()   = ENERGY RELATIVE TO PARENT LEVEL 1 (CM-1)
C                           FOR PARENT LEVEL 'IPA()'
C
C  OUTPUT: (I*4)  IL      = NUMBER OF ENERGY LEVELS (TERMS) OF
C                           RECOMBINED ION
C  OUTPUT: (R*8)  BWNR    = IONISATION POTENTIAL (CM-1) OF LOWEST LEVEL
C                           OF RECOMBINED ION
```

```
C  OUTPUT: (I*4)  IA()    = RECOMBINED ION ENERGY LEVEL INDEX NUMBER
C  OUTPUT: (C*18) CSTRGA()= NOMENCL./CONFIG. FOR RECOMBINED ION LEVEL
C                           'IA()'
C  OUTPUT: (I*4)  ISA()   = MULTIPLICITY FOR RECOMBINED LEVEL 'IA()'
C                           NOTE: (ISA-1)/2 = QUANTUM NUMBER (S)
C  OUTPUT: (I*4)  ILA()   = QUANTUM NUMBER (L) FOR RECOMBINED LEVEL
C                           'IA()'
C  OUTPUT: (R*8)  XJA()   = QUANTUM NUMBER (J) FOR RECOMBINED LEVEL
C                           'IA()'
C                           NOTE: (2*XJA)+1 = STATISTICAL WEIGHT
C  OUTPUT: (R*8)  WA()    = ENERGY RELATIVE TO RECOMBINED LEVEL 1 (CM-1)
C                           FOR RECOMBINED LEVEL 'IA()'
C  OUTPUT: (I*4)  NREP    = NUMBER OF REPRESENTATIVE N-SHELLS
C  OUTPUT: (I*4)  IREPA() = REPRESENTATIVE N-SHELL INDEX NUMBER
C  OUTPUT: (I*4)  NREPA() = REPRESENTATIVE N-SHELLS
C  OUTPUT: (I*4)  IAPRS   = NUMBER OF AUGER RATE INITIAL AND FINAL
C                           PARENT PAIRS
C  OUTPUT: (C*10) CAPRS() = AUGER RATE PARENT PAIR STRING
C                           1ST.DIM: PARENT PAIR INDEX
C  OUTPUT: (I*40) IPAUG(,)= INITIAL AND FINAL PARENTS FOR AUGER BREAKUPS
C                           1ST.DIM: PARENT PAIR INDEX
C                           2ND.DIM: INITIAL AND FINAL PARENT INDICES
C  OUTPUT: (R*8)  AUGA(,) = AUGER RATES (SEC-1)
C                           1ST.DIM: REPRESENTATIVE N-SHELL INDEX
C                           2ND.DIM: PARENT PAIR INDEX
C  OUTPUT: (L*4)  LAUGA(,) = .TRUE. => AUGER RATE PRESENT FOR N-SHELL
C                            .FALSE.=> AUGER RATE NOT PRESENT
C                           1ST.DIM: REPRESENTATIVE N-SHELL INDEX
C                           2ND.DIM: PARENT PAIR INDEX
C  OUTPUT: (I*4)  IPRTI() = INITIAL PARENT BLOCK INDEX
C  OUTPUT: (C*5)  TPRTI() = INITIAL PARENT BLOCK TERM
C  OUTPUT: (I*4)  ISPRTI()= INITIAL PARENT BLOCK SPIN MULTIPLICITY
C  OUTPUT: (R*8)  TEA()   = ELECTRON TEMPERATURES (K)
C  OUTPUT: (R*8)  DIELR(,,)= TERM SELECTIVE DIELEC. COEFFTS.(CM3 S-1)
C                           1ST.DIM: LEVEL INDEX
C                           2ND.DIM: INITIAL PARENT INDEX
C                           3RD.DIM: TEMPERATURE INDEX
C  OUTPUT: (L*4)  LDIELR(,)= .TRUE. => DIEL. PRESENT FOR LEVEL INDEX
C                            .FALSE.=> DIEL. NOT PRESENT FOR LEVEL INDEX
C                           1ST.DIM: LEVEL INDEX
C                           2ND.DIM: INITIAL PARENT INDEX
C  OUTPUT: (I*4)  IPRTF(,) = FINAL PARENT BLOCK INDEX
C  OUTPUT: (C*5)  TPRTF(,) = FINAL PARENT BLOCK TERM
C  OUTPUT: (I*4)  ISPRTF(,)= FINAL PARENT BLOCK SPIN MULTIPLICITY
C  OUTPUT: (I*4)  NSYSF(,) = NO,. OF SPIN SYSTEMS BUILT ON FINAL PARENT
C  OUTPUT: (I*4)  ISYS(,,) = N-SHELL SPIN SYSTEM INDEX FOR FINAL PARENT
C  OUTPUT: (I*4)  ISPSYS(,,)=N-SHELL SPIN SYSTEM FOR FINAL PARENT
C  OUTPUT: (R*8)  DIELN(,,,,) =N-SHELL DIELEC. COEFFTS.(CM3 S-1)
C                           1ST.DIM: REPR. N-SHELL INDEX
C                           2ND.DIM: INITIAL PARENT INDEX
C                           3RD.DIM: FINAL PARENT INDEX
C                           4TH.DIM: SPIN SYSTEM INDEX
C                           5TH.DIM: TEMPERATURE INDEX
C  OUTPUT: (R*8)  LDIELN(,)= .TRUE. => DIEL. PRESENT FOR REPR. N-SHELL
C                            .FALSE.=> DIEL. NOT PRESENT FOR  N-SHELL
C                           1ST.DIM: REPR. N-SHELL INDEX
C                           2ND.DIM: INITIAL PARENT INDEX
C                           3RD.DIM: FINAL PARENT INDEX
C                           4TH.DIM: SPIN SYSTEM INDEX
C  OUTPUT: (R*8)  DIELT(,,,) =N-SHELL DIELEC. COEFFTS.(CM3 S-1)
C                           1ST.DIM: INITIAL PARENT INDEX
```

```
C                           2ND.DIM: FINAL PARENT INDEX
C                           3RD.DIM: SPIN SYSTEM INDEX
C                           4TH.DIM: TEMPERATURE INDEX
C
C          (I*4)  INDX    = GENERAL INDEX
C          (I*4)  INDX1   = GENERAL INDEX
C          (I*4)  II      = GENERAL INDEX
C          (I*4)  I       = GENERAL INDEX
C          (I*4)  IPI     = GENERAL INDEX
C          (I*4)  IPF     = GENERAL INDEX
C          (I*4)  IPFS    = GENERAL INDEX
C          (I*4)  J       = GENERAL INDEX
C          (I*4)  K       = GENERAL INDEX
C
C          (L)    LDATA   = GENERAL READ/DO NOT READ FLAG
C          (L)    LNOPI   = FLAG TO DETERMINE WHETHER HAVE PASSED
C     INTO A NEW INITIAL PARENT BLOCK
C
C          (C*20) C20     = GENERAL CHARACTER STRING
C
C  ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4UNIT     ADAS      FETCH UNIT NUMBER FOR OUTPUT OF MESSAGES
C          I4EIZ0     ADAS      RETURNS NUCL. CHARGE FROM ELEMENT SYMBOL
C          R8FCTN     ADAS      CONVERTS FROM CHARACTER TO REAL VARIABLE
C          XXWORD     ADAS      EXTRACT POSITION OF NUMBER IN BUFFER
C
C  AUTHOR:  H. P. SUMMERS, UNIVERSITY OF STRATHCLYDE
C          JA8.08
C          TEL. 0141-553-4196
C
C  DATE:    05/08/97
C
C  UPDATE: Modified final parent reading block to acount for missing
C          final parents. (do-while to 130 statements).
C          Also added END=999 to read statement to avoid EOF error
C          when there is no data in SYS/SPNSYS block.
C          Martin O'Mullane, 3-10-97
C
C  VERSION: 1.1
C
C  VERSION: 1.2 DATE: 20-02-98
C  MODIFIED: MARTIN O'MULLANE
C     - ERROR IN ASSIGNING NUMBER OF PARENTS IF >= NDPRT.
C  - ADDED TOTAL DR RATE FOR EACH (INITIAL PARENT, FINAL PARENT,
C             SPIN SYSTEM) BLOCK BY SUMMING UP AND INTERPOLATING THE
C             REPRESENTATIVE LEVEL SET.
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      CHARACTER*10        CAPRS(NDAUG)
      CHARACTER*(*)       CSTRGA(NDLEV),            CSTRPA(NDPRT)
      CHARACTER*2         SEQSYM
      CHARACTER*5         TPRTF(NDPRT,NDPRT),       TPRTI(NDPRT)
      INTEGER             IA(NDLEV),   IAPRS,       IL
      INTEGER             ILA(NDLEV),  ILPA(NDPRT), IPA(NDPRT)
      INTEGER             IPAUG(NDAUG,2),           IPRTF(NDPRT,NDPRT)
      INTEGER             IPRTI(NDPRT),             IREPA(NDREP)
      INTEGER             ISA(NDLEV),  ISPA(NDPRT), ISPRTF(NDPRT,NDPRT)
      INTEGER             ISPRTI(NDPRT)
```

```
INTEGER            ISPSYS(NDPRT,NDPRT,2),    ISYS(NDPRT,NDPRT,2)
INTEGER            IUNIT,       IZ,          IZ0,         IZ1
INTEGER            NDAUG,       NDLEV,       NDPRT,       NDREP
INTEGER            NDT,         NPRNT,       NPRNTF,      NPRNTI
INTEGER            NREP,        NREPA(NDREP)
INTEGER            NSYSF(NDPRT,NDPRT),       NTE
LOGICAL            LAUGA(NDREP,NDAUG)
LOGICAL            LDIELN(NDREP,NDPRT,NDPRT,2)
LOGICAL            LDIELR(NDLEV,NDPRT)
REAL*8             AUGA(NDREP,NDAUG),        BWNP,        BWNR
REAL*8             DIELN(NDREP,NDPRT,NDPRT,2,NDT)
REAL*8             DIELR(NDLEV,NDPRT,NDT)
REAL*8             DIELT(NDPRT,NDPRT,2,NDT), TEA(NDT)
REAL*8             WA(NDLEV),   WPA(NDPRT),  XJA(NDLEV)
REAL*8             XJPA(NDPRT)
```

5.70 daspln: Subroutine daspln from library adas4xx

```
      SUBROUTINE DASPLN( NTDIM  , NDTIN  ,
     &                   ITA    , ITVAL  ,
     &                   TFILE  , TEVA   ,
     &                   QDRIN  , QDROUT ,
     &                   LTRNG
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTFILEE: B1SPLN *********************
C
C  PURPOSE:
C         1) PERFORMS CUBIC SPLINE ON LOG(TEMPERATURE) VERSUS LOG(QDRIN)
C            INPUT DATA. ('TFILE' VERSUS 'QDRIN' , ITA DATA PAIRS)
C
C         2) INTERPOLATES 'ITVAL'  QDRIN VALUES USING ABOVE SPLINES AT
C            TEMPERATURES READ IN FROM ISPF PANELS FOR TABULAR OUTPUT.
C            (ANY TEMPERATURE VALUES WHICH REQUIRED EXTRAPOLATION TO
C             TAKE PLACE ARE SET TO ZERO).
C                 - THIS STEP ONLY TAKES PLACE IF 'LOSEL=.TRUE.' -
C
C         3) INTERPOLATES 'ITVAL' QDRIN VALUES USING ABOVE SPLINES AT
C            TEMPERATURES EQUI-DISTANCE ON RANGE OF LOG(TEMPERATURES)
C            STORED IN INPUT 'TFILE' ARRAY.
C
C  CALLING PROGRAM: ADAS201
C
C
C  SUBROUTFILEE:
C
C  INPUT : (I*4)  NTDIM   = MAX. NO. OF TEMPERATURES ALLOWED IN
C                           DATA SET
C  INPUT : (I*4)  NDTIN   = MAX. NO. OF USER TEMPERATURES ALLOWED
C  INPUT : (I*4)  ITA     = INPUT DATA FILE: NUMBER OF DR/TEMPERATURE
C                           PAIRS READ FOR THE TRANSITION BEING ASSESSED
C  INPUT : (I*4)  ITVAL   = NUMBER OF  SPLINE  INTERPOLATED  QDRIN/TEMP.
C                           REQUIRED FOR GRAPHICAL DISPLAY.
C
C  INPUT : (I*4)  TFILE() = INPUT DATA FILE: TEMPERATURES
C  INPUT : (I*4)  TEVA()  = ISPF PANEL ENTERED TEMPERATURES
C
C  INPUT : (R*8)  QDRIN() = INPUT DATA FILE: SELECTED TRANSITION -
C                           QDRIN VALUES AT 'TFILE()'.
C  OUTPUT: (I*4)  QDROUT()= SPLINE INTERPOLATED QDRIN VALUES AT 'TEVA()'
C                           (EXTRAPOLATED VALUES = 0.0).
C
C  OUTPUT: (L*4)  LTRNG() = .TRUE.  => OUTPUT   SPLINE    VALUE    WAS
C                                      INTERPOLATED FOR 'DLOG(TEVA()'.
C                           .FALSE. => OUTPUT   SPLINE    VALUE    WAS
C                                      EXTRAPOLATED FOR 'DLOG(TEVA()'.
C                                      (NOTE: 'YOUT()=0' AS 'IOPT < 0').
C
C          (I*4)  NIN     = PARAMETER = MAX. NO. OF  INPUT  TEMP/QDRIN
C                                       PAIRS MUST BE >= 'ITA'
C          (I*4)  NOUT    = PARAMETER = MAX. NO. OF 'OUTPUT TEMP/QDRIN
C                                       PAIRS MUST BE >= 'ITVAL' & 'ITVAL'
C
C          (I*4)  IARR    = ARRAY SUBSCRIPT USED FOR TEMP/QDRIN PAIRS
C          (I*4)  IOPT    = DEFINES THE BOUNDARY DERIVATIVES FOR THE
C                           SPLINE ROUTFILEE 'XXSPLE', SEE 'XXSPLE'.
```

```
C                           (VALID VALUES = <0, 0, 1, 2, 3, 4)
C
C          (L*4)  LSETX   = .TRUE.  => SET UP SPLINE PARAMETERS RELATFILEG
C                                      TO 'XIN' AXIS.
C                           .FALSE. => DO NOT SET UP SPLINE PARAMETERS
C                                      RELATFILEG TO 'XIN' AXIS.
C                                      (I.E. THEY WERE SET IN A PREVIOUS
C                                            CALL )
C                           (VALUE SET TO .FALSE. BY 'XXSPLE')
C
C          (R*8)  XIN()   = LOG( 'TFILE()' )
C          (R*8)  YIN()   = LOG( 'QDRIN()' )
C          (R*8)  XOUT()  = LOG(TEMPERATURES AT WHICH SPLINES REQUIRED)
C          (R*8)  YOUT()  = LOG(OUTPUT SPLINE INTERPOLATED QDRIN VALUES)
C          (R*8)  DF()    = SPLINE INTERPOLATED DERIVATIVES
C
C          (L*4)  LDUMP() = .TRUE.  => OUTPUT SPLINE VALUE INTRPOLATED
C                                      FOR 'YOUT()'.
C                           .FALSE. => OUTPUT SPLINE VALUE EXTRAPOLATED
C                                      FOR 'YOUT()'.
C                                      (NOTE: USED AS A DUMMY ARGUMENT.
C                                             ALL VALUES WILL BE TRUE.)
C
C NOTE:
C
C ROUTFILEES:
C          ROUTFILEE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          XXSPLE     ADAS      SPLINE SUBROUTFILEE (EXTENDED DIAGNOSTICS)
C          R8FUN1     ADAS      REAL*8 FUNCTION: ( X -> X )
C
C AUTHOR:  Martin O'Mullane  (based on b1spln.for)
C          K1/1/43
C          JET EXT. 5313
C
C
C MODIFIED:
C VERSION:
C
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             ITA,         ITVAL,       NDTIN,       NTDIM
      LOGICAL             LTRNG(NDTIN)
      REAL*8              QDRIN(NTDIM),             QDROUT(NDTIN)
      REAL*8              TEVA(NDTIN), TFILE(NTDIM)
```

5.71 dasumd: Subroutine dasumd from library adas4xx

```
       SUBROUTINE DASUMD ( NDREP  , NDPRT   , NDT    ,
     &                     IPI    , IPF     , IPFS   ,
     &                     MAXTM  , IREPMAX , IREP   , DRMF   ,
     &                     DRMS
     &                   )
C-----------------------------------------------------------------------
C
C ***************** FORTRAN 77 SUBROUTINE: DASUMD  **********************
C
C  VERSION: 1.0
C
C  PURPOSE: TO SUM BADNELL DIELECTRONIC RATE COEFFICIENT DATA OVER THE
C           REPRESENTATIVE SET TO GIVE ZERO DENSITY TOTAL RATE
C           FROM SATELLITE LINES.
C
C           BASED ON B4SUMD
C
C  CALLING PROGRAM: DADATA
C
C
C  INPUT:
C  INPUT :  (I*4) NDREP    = MAXIMUM NUMBER OF REPRESENTATIVE LEVELS
C  INPUT :  (I*4) NDPRT    = MAXIMUM NUMBER OF PARENT STATES
C  INPUT :  (I*4) NDT      = MAXIMUM NUMBER OF TEMPERATURES
C  INPUT :  (I*4) IPI      = INITIAL PARENT
C  INPUT :  (I*4) IPF      = FINAL PARENT
C  INPUT :  (I*4) IPFS     = SOIN SYSTEM
C  INPUT :  (I*4) DRMF(,,) = BADNELL DIELECTRONIC DATA (CM3 S-1)
C                            1ST.DIM: REPR. N-SHELL INDEX
C                            2ND.DIM: INITIAL PARENT INDEX
C                            3RD.DIM: FINAL PARENT INDEX
C                            4TH.DIM: SPIN SYSTEM INDEX
C                            5TH.DIM: TEMPERATURE INDEX
C  INPUT :  (I*4) NBT      = NO. OF TEMPERATURES
C  INPUT :  (I*4) IREPMAX  = NO OF REPRESENTATIVE LEVELS
C  INPUT :  (I*4) IREP()   = SET OF REPRESENTATIVE LEVELS
C
C  OUTPUT:  (R*8) DRMS()   = SUMMED DR RATE COEFFICIENTS (CM3 S-1)
C                            1ST.DIM: INITIAL PARENT INDEX
C                            2ND.DIM: FINAL PARENT INDEX
C                            3RD.DIM: SPIN SYSTEM INDEX
C                            4TH.DIM: TEMPERATURE INDEX
C
C           (I*4) NREP     = GENERAL LEVEL INDEX
C           (I*4) IN       = GENERAL INDEX
C           (I*4) IT       = GENERAL INDEX
C           (R*8) V        = GENERAL VARIABLE FOR N-SHELL
C           (R*8) V1       = GENERAL VARIABLE FOR N-SHELL
C           (R*8) Y        = GENERAL VARIABLE FOR N-SHELL
C           (R*8) Y0       = GENERAL VARIABLE FOR N-SHELL
C           (R*8) Y1       = GENERAL VARIABLE FOR N-SHELL
C
C
C  AUTHOR:   Martin O'Mullane
C            JET
C
C  DATE:     19-02-98
C
C  VERSION:  1.1 DATE: 03-03-98
```

```
C  MODIFIED: RICHARD MARTIN
C - PUT THROUGH SCCS.
C
C-----------------------------------------------------------------
      INTEGER             IPF,          IPFS,         IPI
      INTEGER             IREP(NDREP), IREPMAX,     MAXTM,       NDPRT
      INTEGER             NDREP,       NDT
      REAL*8              DRMF(NDREP,NDPRT,NDPRT,2,NDT)
      REAL*8              DRMS(NDPRT,NDPRT,2,NDT)
```

5.72 datitl: Subroutine datitl from library adas4xx

```
      SUBROUTINE DATITL( IOPT     ,
     &             IP_RES   , IL_RES ,
     &   DSFULL   ,
     &                   TITLX
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: DATITL *********************
C
C  PURPOSE:  TO CREATE THE DESCRIPTIVE TITLE FOR SELECTED DATA-BLOCK.
C
C  CALLING PROGRAM: ADAS410
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  IOPT       = Type of plot
C  INPUT : (I*4)  IL_RES     = resolved DR parent
C  INPUT : (I*4)  IL_RES     = resolved DR level
C  I
C  INPUT : (C*80) DSFULL   = FULL INPUT DATA SET NAME
C
C
C  INPUT : (C*9)  CDONOR   = SELECTED DATA-BLOCK: DONOR IDENTITY
C
C  INPUT : (C*9)  CRECVR   = SELECTED DATA-BLOCK: RECEIVER IDENTITY
C
C  INPUT : (C*10) CFSTAT  = SELECTED DATA-BLOCK: FINAL STATE SPEC.
C
C  OUTPUT: (C*120) TITLX    = SELECTED DATA-BLOCK: DESCRIPTIVE TITLE
C
C          (C*2)  C2       = GENERAL USE 2 BYTE  CHARACTER  STRING
C
C
C AUTHOR   : Martin O'Mullane
C            K1/1/43
C            JET EXT. 5313
C
C DATE     : 25/9/97
C
C VERSION  :
C MODIFIED :
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      CHARACTER*120       DSFULL,      TITLX
      INTEGER             IL_RES,      IOPT,        IP_RES
```

5.73 dbspln: Subroutine dbspln from library adas4xx

```
      SUBROUTINE DBSPLN( NTDIM  , NDTIN  ,
     &                   ITA    , ITVAL  ,
     &                   TFILE  , TEVA   ,
     &                   QDRIN  , QDROUT ,
     &                   LTRNG
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTFILEE: DBSPLN *********************
C
C  PURPOSE:
C         1) PERFORMS CUBIC SPLINE ON LOG(TEMPERATURE) VERSUS LOG(QDRIN)
C            INPUT DATA. ('TFILE' VERSUS 'QDRIN' , ITA DATA PAIRS)
C
C         2) INTERPOLATES 'ITVAL'  QDRIN VALUES USING ABOVE SPLINES AT
C            TEMPERATURES READ IN FROM ISPF PANELS FOR TABULAR OUTPUT.
C            (ANY TEMPERATURE VALUES WHICH REQUIRED EXTRAPOLATION TO
C             TAKE PLACE ARE SET TO ZERO).
C
C         3) INTERPOLATES 'ITVAL' QDRIN VALUES USING ABOVE SPLINES AT
C            TEMPERATURES EQUI-DISTANCE ON RANGE OF LOG(TEMPERATURES)
C            STORED IN INPUT 'TFILE' ARRAY.
C
C  CALLING PROGRAM: ADAS411
C
C
C  SUBROUTFILEE:
C
C  INPUT : (I*4)  NTDIM   = MAX. NO. OF TEMPERATURES ALLOWED IN
C                           DATA SET
C  INPUT : (I*4)  NDTIN   = MAX. NO. OF USER TEMPERATURES ALLOWED
C  INPUT : (I*4)  ITA     = INPUT DATA FILE: NUMBER OF DR/TEMPERATURE
C                           PAIRS READ FOR THE TRANSITION BEING ASSESSED
C  INPUT : (I*4)  ITVAL   = NUMBER OF  SPLINE  INTERPOLATED  QDRIN/TEMP.
C                           REQUIRED FOR GRAPHICAL DISPLAY.
C
C  INPUT : (I*4)  TFILE() = INPUT DATA FILE: TEMPERATURES
C  INPUT : (I*4)  TEVA()  = ISPF PANEL ENTERED TEMPERATURES
C
C  INPUT : (R*8)  QDRIN() = INPUT DATA FILE: SELECTED TRANSITION -
C                           QDRIN VALUES AT 'TFILE()'.
C  OUTPUT: (I*4)  QDROUT()= SPLINE INTERPOLATED QDRIN VALUES AT 'TEVA()'
C                           (EXTRAPOLATED VALUES = 0.0).
C
C  OUTPUT: (L*4)  LTRNG() = .TRUE.  => OUTPUT   SPLINE    VALUE    WAS
C                                      INTERPOLATED FOR 'DLOG(TEVA()'.
C                           .FALSE. => OUTPUT   SPLINE    VALUE    WAS
C                                      EXTRAPOLATED FOR 'DLOG(TEVA()'.
C                                      (NOTE: 'YOUT()=0' AS 'IOPT < 0').
C
C          (I*4)  NIN     = PARAMETER = MAX. NO. OF  INPUT  TEMP/QDRIN
C                                      PAIRS MUST BE >= 'ITA'
C          (I*4)  NOUT    = PARAMETER = MAX. NO. OF 'OUTPUT TEMP/QDRIN
C                                      PAIRS MUST BE >= 'ITVAL' & 'ITVAL'
C
C          (I*4)  IARR    = ARRAY SUBSCRIPT USED FOR TEMP/QDRIN PAIRS
C          (I*4)  IOPT    = DEFINES THE BOUNDARY DERIVATIVES FOR THE
C                           SPLINE ROUTFILEE 'XXSPLE', SEE 'XXSPLE'.
C                           (VALID VALUES = <0, 0, 1, 2, 3, 4)
```

```
C
C          (L*4)  LSETX   = .TRUE.  => SET UP SPLINE PARAMETERS RELATFILEG
C                                     TO 'XIN' AXIS.
C                           .FALSE. => DO NOT SET UP SPLINE PARAMETERS
C                                     RELATFILEG TO 'XIN' AXIS.
C                                     (I.E. THEY WERE SET IN A PREVIOUS
C                                           CALL )
C                           (VALUE SET TO .FALSE. BY 'XXSPLE')
C
C          (R*8)  XIN()   = LOG( 'TFILE()' )
C          (R*8)  YIN()   = LOG( 'QDRIN()' )
C          (R*8)  XOUT()  = LOG(TEMPERATURES AT WHICH SPLINES REQUIRED)
C          (R*8)  YOUT()  = LOG(OUTPUT SPLINE INTERPOLATED QDRIN VALUES)
C          (R*8)  DF()    = SPLINE INTERPOLATED DERIVATIVES
C
C
C NOTE:
C
C ROUTFILEES:
C          ROUTFILEE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          XXSPLE     ADAS      SPLINE SUBROUTFILEE (EXTENDED DIAGNOSTICS)
C          R8FUN1     ADAS      REAL*8 FUNCTION: ( X -> X )
C
C AUTHOR:  H. P. Summers
C          JA8.08
C          TEL. 0141-553-4196
C
C
C DATE:    10/11/97
C
C UPDATE:
C
C VERSION: 1.1 DATE: 10-03-98
C MODIFIED: RICHARD MARTIN
C - PUT UNDER SCCS CONTROL
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             ITA,         ITVAL,       NDTIN,       NTDIM
      LOGICAL             LTRNG(NDTIN)
      REAL*8              QDRIN(NTDIM),             QDROUT(NDTIN)
      REAL*8              TEVA(NDTIN), TFILE(NTDIM)
```

5.74 dclnorm: Subroutine dclnorm from library adas4xx

```
      SUBROUTINE DCLNORM( NDLEV  , NDMET  ,
     &                    NORD   ,
     &                    STCK   ,
     &                    COEF
     &                  )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: DCLNORM *********************
C
C  PURPOSE: TO NORMALISE LINE EMISSIVITY
C           ADAPTED FROM B6NORM
C
C  CALLING PROGRAM:  XCOEF
C
C  SUBROUTINE:
C
C  INPUT :  (I*4)  NDLEV   = MAXIMUM NUMBER OF ENERGY LEVELS ALLOWED
C  INPUT :  (I*4)  NDMET   = MAXIMUM NUMBER OF METASTABLE LEVELS ALLOWED
C
C  INPUT :  (I*4)  NORD    = NUMBER OF ORDINARY EXCITED LEVELS
C
C  INPUT :  (R*8)  STCK(,) = POPULATION MATRIX COVERING ALL NON-METAST-
C                            ABLE/ORDINARY EXCITED LEVELS AS FUNCTION
C                            OF METASTABLE INDEX.
C                            VALUES FOR GIVEN TEMPERATURE AND DENSITY.
C                            1st DIMENSION: ORDINARY EXCITED LEVEL INDEX
C                            2nd DIMENSION: METASTABLE LEVEL INDEX
C
C  I/O   :  (R*8)  COEF    = INPUT:
C                            LINE EMISSIVITY
C                               A(J->K) * [N(J)/N(1)]
C                            AT FIXED TEMPERATURE AND DENSITY.
C                            (UNITS: ERGS CM3 SEC-1)    >>>>?<<<<
C                            OUTPUT:
C                            NORMALISED TO TOTAL STAGE POPULATION
C                               [N(1)/SUM(N(I)] * A(J->K) * [N(J)/N(1)]
C
C           (I*4)  IS1     = ORDINARY EXCITED LEVEL INDEX
C
C           (R*8)  STOTX   = VARIABLE USED TO SUM STAGE TOTAL POPULATN.
C                            (INITIAL VALUE = 1 => GROUND)
C
C ROUTINES: NONE
C
C NOTE:
C
C AUTHOR:  A. Lanzafame, University of Strathclyde
C
C DATE:    apr28-95
C
C UPDATE:
C
C VERSION 1.1 DATE: 27-10-97
C RICHARD MARTIN.
C PUT UNDER SCCS CONTROL.
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
```

```
INTEGER             NDLEV,       NDMET,       NORD
REAL*8              COEF,        STCK(NDLEV,NDMET)
```

5.75 dcpopm: Subroutine dcpopm from library adas4xx

```
      SUBROUTINE DCPOPM( NDTEM  , NDMET , NDLEV ,
     &                   MAXT   , NMET  ,
     &                            DENSA , IMETR ,
     &                            LRSEL , LHSEL ,
     &                            RATIA , RATHA ,
     &                   STCKM  , STVRM , STVHM ,
     &                   POPAR
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: DCPOPM *********************
C
C  PURPOSE: TO CONSTRUCT METASTABLE LEVEL POPULATIONS.
C
C  CALLING PROGRAM:  XCOEF
C
C  SUBROUTINE:
C
C  INPUT :  (I*4)  NDTEM   = MAXIMUM NUMBER OF TEMP/DENS PAIRS ALLOWED
C  INPUT :  (I*4)  NDMET   = MAXIMUM NUMBER OF METASTABLE LEVELS ALLOWED
C  INPUT :  (I*4)  NDLEV   = MAXIMUM NUMBER OF ENERGY LEVELS ALLOWED
C
C  INPUT :  (I*4)  MAXT    = NO. INPUT TEMP/DENSITY PAIRS ( 1 ->'NDTEM')
C  INPUT :  (I*4)  NMET    = NUMBER OF METASTABLES LEVELS ( 1 ->'NDMET')
C
C  INPUT :  (R*8)  DENSA() = ELECTRON DENSITIES  (UNITS: CM-3)
C  INPUT :  (I*4)  IMETR() = INDEX OF METASTABLE IN COMPLETE LEVEL LIST
C                            (ARRAY SIZE = 'NDMET' )
C
C  INPUT :  (L*4)  LRSEL   = .TRUE.  => FREE ELECTRON RECOMBINATION
C                                       REQUESTED.
C                          = .FALSE. => FREE ELECTRON RECOMBINATION
C                                       NOT REQUESTED.
C  INPUT :  (L*4)  LHSEL   = .TRUE.  => CHARGE TRANSFER FROM NEUTRAL
C                                       HYDROGREN REQUESTED.
C                          = .FALSE. => CHARGE TRANSFER FROM NEUTRAL
C                                       HYDROGREN NOT REQUESTED.
C
C  INPUT :  (R*8)  RATIA() = RATIO ( N(Z+1)/N(Z)  STAGE ABUNDANCIES )
C  INPUT :  (R*8)  RATHA() = RATIO (NEUTRAL H DENSITY/ELECTRON DENSITY)
C
C  INPUT :  (R*8)  STCKM(,) = METASTABLE POPULATIONS STACK:
C                               1st DIMENSION: METASTABLE INDEX
C                               2nd DIMENSION: TEMPERATURE INDEX
C  INPUT :  (R*8)  STVRM(,) = METASTABLE LEVEL:
C                             FREE-ELECTRON RECOMBINATION COEFFICIENTS
C                             (UNITS* CM**3/SEC-1)
C                              1st DIMENSION: METASTABLE INDEX
C                              2nd DIMENSION: TEMPERATURE INDEX
C  INPUT :  (R*8)  STVHM(,) = METASTABLE LEVEL:
C                             CHARGE-EXCHANGE RECOMBINATION COEFFICIENTS
C                             (UNITS* CM**3/SEC-1)
C                              1st DIMENSION: METASTABLE INDEX
C                              2nd DIMENSION: TEMPERATURE INDEX
C
C  OUTPUT:  (R*8)  POPAR(,) = LEVEL POPULATIONS
C                              1st DIMENSION: LEVEL INDEX
C                              2nd DIMENSION: TEMPERATURE INDEX
C                              (ON OUTPUT CONTAINS POPULATIONS FOR
```

```
C                                METASTABLE LEVELS ONLY.)
C
C           (R*8) DCOEF     = DENSITY MULTIPLIED BY RELEVANT RATIOS  FOR
C                             CALCULATING RECOMBINATION CONTRIBUTIONS.
C
C           (I*4) IT        = TEMPERATURE ARRAY INDEX
C           (I*4) IM        = METASTABLE LEVEL ARRAY INDEX
C
C
C ROUTINES: NONE
C
C AUTHOR:  HP SUMMERS
C          K1/1/57
C          JET EXT. 4941
C
C DATE:    27/ 6/91
C
C VERSION: 1.1 RICHARD MARTIN DATE: 27-10-97
C PUT UNDER SCCS CONTROL.
C NAME CHANGED FROM BHPOPM TO DCPOPM
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IMETR(NDMET),             MAXT,        NDLEV
      INTEGER             NDMET,       NDTEM,       NMET
      LOGICAL             LHSEL,       LRSEL
      REAL*8              DENSA(NDTEM),             POPAR(NDLEV,NDTEM)
      REAL*8              RATHA(NDTEM),             RATIA(NDTEM)
      REAL*8              STCKM(NDMET,NDTEM),       STVHM(NDMET,NDTEM)
      REAL*8              STVRM(NDMET,NDTEM)
```

5.76 dcpopo: Subroutine dcpopo from library adas4xx

```
      SUBROUTINE DCPOPO( NDTEM  , NDMET , NDLEV ,
     &                   MAXT   , NMET  , NORD  ,
     &                            DENSA , IMETR , IORDR ,
     &                            LRSEL , LHSEL ,
     &                            RATIA , RATHA ,
     &                   STACK  , STVR  , STVH  ,
     &                   POPAR
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: DCPOPO *********************
C
C  PURPOSE: TO CONSTRUCT ORDINARY/NON-METASTABLE LEVEL POPULATIONS.
C
C  CALLING PROGRAM:  XCOEF
C
C  SUBROUTINE:
C
C  INPUT :  (I*4)  NDTEM   = MAXIMUM NUMBER OF TEMP/DENS PAIRS ALLOWED
C  INPUT :  (I*4)  NDMET   = MAXIMUM NUMBER OF METASTABLE LEVELS ALLOWED
C  INPUT :  (I*4)  NDLEV   = MAXIMUM NUMBER OF ENERGY LEVELS ALLOWED
C
C  INPUT :  (I*4)  MAXT    = NO. OF INPUT TEMP/DENS PAIRS ( 1 ->'NDTEM')
C  INPUT :  (I*4)  NMET    = NUMBER OF METASTABLES LEVELS ( 1 ->'NDMET')
C  INPUT :  (I*4)  NORD    = NUMBER OF ORDINARY LEVELS ( 1 ->'NDLEV')
C
C  INPUT :  (R*8)  DENSA() = ELECTRON DENSITIES  (UNITS: CM-3)
C  INPUT :  (I*4)  IMETR() = INDEX OF METASTABLE IN COMPLETE LEVEL LIST
C                            (ARRAY SIZE = 'NDMET' )
C  INPUT :  (I*4)  IORDR() =INDEX OF ORDINARY EXCITED LEVELS IN COMPLETE
C                            LEVEL LIST.
C                            (ARRAY SIZE = 'NDLEV' )
C
C  INPUT :  (L*4)  LRSEL   = .TRUE.  => FREE ELECTRON RECOMBINATION
C                                       REQUESTED.
C                          = .FALSE. => FREE ELECTRON RECOMBINATION
C                                       NOT REQUESTED.
C  INPUT :  (L*4)  LHSEL   = .TRUE.  => CHARGE TRANSFER FROM NEUTRAL
C                                       HYDROGREN REQUESTED.
C                          = .FALSE. => CHARGE TRANSFER FROM NEUTRAL
C                                       HYDROGREN NOT REQUESTED.
C
C  INPUT :  (R*8)  RATIA() = RATIO ( N(Z+1)/N(Z)  STAGE ABUNDANCIES )
C  INPUT :  (R*8)  RATHA() = RATIO (NEUTRAL H DENSITY/ELECTRON DENSITY)
C
C  INPUT : (R*8) STACK(,,) = ORDINARY EXCITED LEVEL POPULAT'N DEPENDENCE
C                            ON METASTABLE LEVEL.
C                             1st DIMENSION: ORDINARY LEVEL INDEX
C                             2nd DIMENSION: METASTABLE INDEX
C                             3rd DIMENSION: TEMPERATURE INDEX
C  INPUT : (R*8) STVR(,)   = ORDINARY EXCITED LEVEL:
C                             FREE-ELECTRON RECOMBINATION COEFFICIENTS
C                             (UNITS* CM**3/SEC-1)
C                             1st DIMENSION: ORDINARY LEVEL INDEX
C                             2nd DIMENSION: TEMPERATURE INDEX
C  INPUT : (R*8) STVH(,)   = ORDINARY EXCITED LEVEL:
C                             CHARGE-EXCHANGE RECOMBINATION COEFFICIENTS
C                             (UNITS* CM**3/SEC-1)
C                             1st DIMENSION: ORDINARY LEVEL INDEX
```

```
C                              2nd DIMENSION: TEMPERATURE INDEX
C
C  I/O   :  (R*8)  POPAR(,) = LEVEL POPULATIONS
C                               1st DIMENSION: LEVEL INDEX
C                               2nd DIMENSION: TEMPERATURE INDEX
C                                ON INPUT : CONTAINS POPULATIONS FOR
C                                           METASTABLE LEVELS ONLY.
C                                ON OUTPUT: CONTAINS POPULATIONS FOR
C                                           ALL LEVELS.
C
C           (R*8) DCOEF     = DENSITY MULTIPLIED BY RELEVANT RATIOS  FOR
C                             CALCULATING RECOMBINATION CONTRIBUTIONS.
C
C           (I*4) IT        = TEMPERATURE ARRAY INDEX
C           (I*4) IN        = DENSITY ARRAY INDEX
C           (I*4) IO        = ORDINARY LEVEL ARRAY INDEX
C           (I*4) IM        = METASTABLE LEVEL ARRAY INDEX
C
C
C ROUTINES: NONE
C
C AUTHOR:  HP SUMMERS
C          K1/1/57
C          JET EXT. 4941
C
C DATE:    27/06/91
C

C VERSION 1.1  RICHARD MARTIN DATE: 27-10-97
C PUT UNDER SCCS CONTROL.
C NAME CHANGED FROM BHPOPO TO DCPOPO
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IMETR(NDMET),              IORDR(NDLEV)
      INTEGER             MAXT,        NDLEV,        NDMET,       NDTEM
      INTEGER             NMET,        NORD
      LOGICAL             LHSEL,       LRSEL
      REAL*8              DENSA(NDTEM),              POPAR(NDLEV,NDTEM)
      REAL*8              RATHA(NDTEM),              RATIA(NDTEM)
      REAL*8              STACK(NDLEV,NDMET,NDTEM), STVH(NDLEV,NDTEM)
      REAL*8              STVR(NDLEV,NDTEM)
```

5.77 dcstka: Subroutine dcstka from library adas4xx

```
      SUBROUTINE DCSTKA( NDLEV  , NDMET  ,
     &                   NORD   , NMET   ,
     &                   IORDR  , IMETR  ,
     &                   CMAT   , CC     ,
     &                   STCK
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: DCSTKA *********************
C
C  PURPOSE: TO STACK UP IN  'STCK'  THE NON-METASTABLE/ORDINARY EXCITED
C           LEVEL POPULATION DEPENDENCE ON METASTABLE LEVEL FOR A GIVEN
C           TEMPERATURE AND DENSITY.
C
C  CALLING PROGRAM:  XCOEF
C
C  SUBROUTINE:
C
C  INPUT :  (I*4)  NDLEV   = MAXIMUM NUMBER OF ENERGY LEVELS ALLOWED
C  INPUT :  (I*4)  NDMET   = MAXIMUM NUMBER OF METASTABLE LEVELS ALLOWED
C
C  INPUT :  (I*4)  NORD    = NUMBER OF ORDINARY EXCITED LEVELS
C  INPUT :  (I*4)  NMET    = NUMBER OF METASTABLE LEVELS
C
C  INPUT :  (I*4)  IMETR() = INDEX OF METASTABLE IN COMPLETE LEVEL LIST
C                            (ARRAY SIZE = 'NDMET' )
C  INPUT :  (I*4)  IORDR() =INDEX OF ORDINARY EXCITED LEVELS IN COMPLETE
C                           LEVEL LIST.
C                           (ARRAY SIZE = 'NDLEV' )
C
C  INPUT :  (R*8)  CMAT(,) = INVERTED   RATE   MATRIX   COVERING   ALL
C                           NON-METASTABLE/ORDINARY EXCITED LEVELS.
C                           (UNITS: SEC)
C                           VALUES FOR GIVEN TEMPERATURE AND DENSITY.
C                           1st DIMENSION: ORDINARY EXCITED LEVEL INDEX
C                           2nd DIMENSION: ORDINARY EXCITED LEVEL INDEX
C
C  INPUT :  (R*8)  CC(,)   = RATE MATRIX COVERING ALL TRANSITIONS
C                           (UNITS: SEC-1)
C                           VALUES FOR GIVEN TEMPERATURE AND DENSITY.
C                           1st DIMENSION: ENERGY LEVEL INDEX
C                           2nd DIMENSION: ENERGY LEVEL INDEX
C
C  OUTPUT:  (R*8)  STCK(,) = POPULATION MATRIX COVERING ALL NON-METAST-
C                           ABLE/ORDINARY EXCITED LEVELS AS FUNCTION
C                           OF METASTABLE INDEX.
C                           VALUES FOR GIVEN TEMPERATURE AND DENSITY.
C                           1st DIMENSION: ORDINARY EXCITED LEVEL INDEX
C                           2nd DIMENSION: METASTABLE LEVEL INDEX
C
C           (I*4)  IS1     = ORDINARY EXCITED LEVEL INDEX
C           (I*4)  IS2     = ORDINARY EXCITED LEVEL INDEX
C           (I*4)  IM      = METASTABLE LEVEL ARRAY INDEX
C
C           (R*8)  POP     = VARIABLE USED TO SUM POPULATION VALUES
C
C
C ROUTINES: NONE
C
```

```
C NOTE:
C        IF:     n  =  number of ordinary/non-metastable levels
C                m  =  number of metastable levels
C           Ro(nxn) = Rate matrix (sec-1) covering transistions between
C                     all possible pairs of ordinary levels.
C                     row   : final   level
C                     column: initial level
C                     (Inverse Ro-1(nxn) = 'CMAT(,)' )
C           Rm(nxm) = Rate matrix (sec-1) covering transistions between
C                     all combinations of ordinary and metastable level
C                     ( = 'CC(,)' - ordinary level part )
C           P(nxm)  = Population matrix giving the population dependence
C                     of each ordinary level on metastable level.
C                     ( = 'STCK(,)' )
C
C           Therefore:  Ro(nxn).P(nxm) = Rm(nxm)
C
C            =>          P(nxm)  = Ro-1(nxn).Rm(nxm)
C
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 4569
C
C DATE:    09/10/90
C
C UPDATE:  20/05/93 - P BRIDEN: STCK ARRAY CHANGED FROM REAL*8 -> REAL*4
C
C          Alessandro Lanzafame, STCK back to real*8
C
C VERSION 1.1  DATE: 27-10-97
C RICHARD MARTIN
C PUT UNDER SCCS CONTROL
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IMETR(NDMET),             IORDR(NDLEV)
      INTEGER             NDLEV,       NDMET,       NMET,        NORD
      REAL*8              CC(NDLEV,NDLEV),          CMAT(NDLEV,NDLEV)
      REAL*8              STCK(NDLEV,NDMET)
```

5.78 dcstkc: Subroutine dcstkc from library adas4xx

```
      SUBROUTINE DCSTKC( NDLEV  , NDMET  ,
     &                   NORD   , NMET   ,
     &                   IORDR  , IMETR  ,
     &                   CC     , STCK   ,
     &                   CRED
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: DCSTKC *********************
C
C  PURPOSE: TO STACK UP IN  'CRED'  THE TRANSITION RATE BETWEEN METASTA-
C           BLE LEVELS FOR A GIVEN TEMPERATURE STABLE LEVEL FOR A GIVEN
C           TEMPERATURE AND DENSITY.
C
C  CALLING PROGRAM:  XCOEF
C
C  SUBROUTINE:
C
C  INPUT :  (I*4)  NDLEV   = MAXIMUM NUMBER OF ENERGY LEVELS ALLOWED
C  INPUT :  (I*4)  NDMET   = MAXIMUM NUMBER OF METASTABLE LEVELS ALLOWED
C
C  INPUT :  (I*4)  NORD    = NUMBER OF ORDINARY EXCITED LEVELS
C  INPUT :  (I*4)  NMET    = NUMBER OF METASTABLE LEVELS
C
C  INPUT :  (I*4)  IMETR() = INDEX OF METASTABLE IN COMPLETE LEVEL LIST
C                            (ARRAY SIZE = 'NDMET' )
C  INPUT :  (I*4)  IORDR() =INDEX OF ORDINARY EXCITED LEVELS IN COMPLETE
C                           LEVEL LIST.
C                           (ARRAY SIZE = 'NDLEV' )
C
C  INPUT :  (R*8)  CC(,)   = RATE MATRIX COVERING ALL TRANSITIONS
C                           (UNITS: SEC-1)
C                           VALUES FOR GIVEN TEMPERATURE AND DENSITY.
C                           1st DIMENSION: ENERGY LEVEL INDEX
C                           2nd DIMENSION: ENERGY LEVEL INDEX
C  INPUT :  (R*8)  STCK(,) = POPULATION MATRIX COVERING ALL NON-METAST-
C                           ABLE/ORDINARY EXCITED LEVELS AS FUNCTION
C                           OF METASTABLE INDEX.
C                           VALUES FOR GIVEN TEMPERATURE AND DENSITY.
C                           1st DIMENSION: ORDINARY EXCITED LEVEL INDEX
C                           2nd DIMENSION: METASTABLE LEVEL INDEX
C
C  OUTPUT:  (R*8)  CRED(,) = MATRIX  OF  TRANSITION   RATES   BETWEEN
C                           METASTABLE LEVELS.
C                           (UNITS: SEC-1)
C                           VALUES FOR GIVEN TEMPERATURE AND DENSITY.
C                           1st DIMENSION: METASTABLE LEVEL INDEX
C                           2nd DIMENSION: METASTABLE LEVEL INDEX
C
C           (I*4)  IM1     = METASTABLE LEVEL ARRAY INDEX
C           (I*4)  IM2     = METASTABLE LEVEL ARRAY INDEX
C           (I*4)  IS      = ORDINARY EXCITED LEVEL INDEX
C
C
C ROUTINES: NONE
C
C NOTE:
C           CRED(IM1,IM2) = ( the transition rate from IM2 to IM1 )
C                                                +
```

```
C                           SUM( (the transistion rate from ordinary
C                                 level IS to IM1)  x (the population
C                                 in metastable level IM2 that excite
C                                 to oridinary level IS) )
C
C                           ABOVE SUM IS OVER ALL ORDINARY LEVELS.
C
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 4569
C
C DATE:    09/10/90
C
C UPDATE:  20/05/93 - P BRIDEN: STCK ARRAY CHANGED FROM REAL*8 -> REAL*4
C
C          apr28-95   A. Lanzafame STCK array back to real*8
C
C VERSION 1.1 DATE: 27-10-97
C RICHARD MARTIN
C PUT UNDER SCCS CONTROL
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IMETR(NDMET),             IORDR(NDLEV)
      INTEGER             NDLEV,       NDMET,       NMET,        NORD
      REAL*8              CC(NDLEV,NDLEV),          CRED(NDMET,NDMET)
      REAL*8              STCK(NDLEV,NDMET)
```

5.79 dddata: Subroutine dddata from library adas4xx

```
      SUBROUTINE  DDDATA( IUNIT  , NDPRT  , NDPRTI, NDREP , NDLEV ,
     &                    NDMET  , NDAUG  , NDT   ,
     &                    SEQSYM , IZ     , IZ0   , IZ1   ,
     &                    NPRF   , BWNF   , NPRFM , IPRFM ,
     &                    NPRI   , IPRI   ,
     &                    IPA    , CSTRPA , ISPA  , ILPA  , XJPA  ,
     &                    WPA    ,
     &                    NLEV   , BWNI   , NLEVM , ILEVM ,
     &                    IA     , CSTRGA , ISA   , ILA   , XJA   ,
     &                    WA     ,
     &                    LRAUG  , NPF    , INDF  , RAUG  ,
     &                    IMETI  , LSYSM  , NSYSM , LREPM , NREPM ,
     &                    LIONIS , LRION  , RION  ,
     &                    LEXCIT , LREXC  , REXC  ,
     &                    NPIS   , IPRTI  , ISYSI , ISPSYI,
     &                    IMETF  , NVALS  , NREPI , IREP  ,
     &                    AUGN   , LEXCN  , EXCN  ,
     &                    NTE    , TEA
     &                  )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: DDDATA *********************
C
C  PURPOSE:  TO FETCH DATA FROM INPUT ADF23 DATA SET.
C
C  CALLING PROGRAM: ADAS213/ADAS413
C
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  IUNIT   = UNIT TO WHICH INPUT FILE IS ALLOCATED
C
C  INPUT : (I*4)  NDPRT   = MAXIMUM NUMBER OF PARENT STATES
C  INPUT : (I*4)  NDPRTI  = MAXIMUM NUMBER OF INTERMEDIATE PARENT STATES
C  INPUT : (I*4)  NDREP   = MAX. NUMBER OF REPRESENTATIVE N-SHELLS
C  INPUT : (I*4)  NDLEV   = MAXIMUM NUMBER OF RESOLVED LEVELS
C  INPUT : (I*4)  NDMET   = MAXIMUM NUMBER OF IONISING ION METASTABLES
C  INPUT : (I*4)  NDAUG   = MAXIMUM NUMBER OF AUGER RATE INITIAL AND
C                           FINAL PARENT PAIRS
C  INPUT : (I*4)  NDT     = MAX. NUMBER OF ELECTRON TEMPERATURES
C
C  OUTPUT: (C*2)  SEQSYM  = RECOMBINED ION SEQ
C  OUTPUT: (I*4)  IZ      = RECOMBINED ION CHARGE
C  OUTPUT: (I*4)  IZ0     = NUCLEAR CHARGE
C  OUTPUT: (I*4)  IZ1     = RECOMBINING ION CHARGE
C  OUTPUT: (I*4)  NPRF    = NUMBER OF FINAL PARENTS
C  OUTPUT: (R*8)  BWNF    = BINDING WAVE NO. OF GROUND PARENT (CM-1)
C  OUTPUT: (I*4)  NPRFM   = NUMBER OF FINAL PARENTS WHICH ARE METASTABLES
C  OUTPUT: (I*4)  IPRFM() = CROSS-REFERENCING OF FINAL METASTABLE
C                           PARENTS TO FINAL PARENT LIST.
C  OUTPUT: (I*4)  NPRI    = NUMBER OF FINAL PARENTS WHICH ARE INTERMEDIATE
C                           PARENTS FOR REPR. N-SHELL DOUBLY EXCITED STATES
C  OUTPUT: (I*4)  IPRI()  = CROSS-REFERENCING OF INTERMEDIATE
C                           PARENTS TO FINAL PARENT LIST.
C  OUTPUT: (I*4)  IPA()   = INDEX OF FINAL PARENT ENERGY LEVELS
C  OUTPUT: (C*18) CSTRPA()= NOMENCL./CONFIG. FOR PARENT LEVEL 'IPA()'
C  OUTPUT: (I*4)  ISPA()  = MULTIPLICITY FOR PARENT LEVEL 'IPA()'
C                           NOTE: (ISPA-1)/2 = QUANTUM NUMBER (SP)
C  OUTPUT: (I*4)  ILPA()  = QUANTUM NUMBER (LP) FOR PARENT LEVEL 'IPA()'
```

```
C  OUTPUT: (R*8)  XJPA()  = QUANTUM NUMBER (JP) FOR PARENT LEVEL 'IPA()'
C                           NOTE: (2*XJPA)+1 = STATISTICAL WEIGHT
C  OUTPUT: (R*8)  WPA()   = ENERGY RELATIVE TO PARENT LEVEL 1 (CM-1)
C                           FOR PARENT LEVEL 'IPA()'
C
C  OUTPUT: (I*4)  NLEV    = NUMBER OF ENERGY LEVELS (TERMS) OF THE
C                           IONISING ION
C  OUTPUT: (R*8)  BWNI    = IONISATION POTENTIAL (CM-1) OF LOWEST LEVEL
C                           OF IONISING ION
C  OUTPUT: (I*4)  NLEVM   = NUMBER OF IONISING ION LEVELS WHICH ARE METASTABLES
C  OUTPUT: (I*4)  ILEVM() = CROSS-REFERENCING OF IONISNG ION METASTABLES
C                           TO IONISING ION LEVEL LIST.
C  OUTPUT: (I*4)  IA()    = IONISING ION ENERGY LEVEL INDEX NUMBER
C  OUTPUT: (C*18) CSTRGA()= NOMENCL./CONFIG. FOR RECOMBINED ION LEVEL
C                           'IA()'
C  OUTPUT: (I*4)  ISA()   = MULTIPLICITY FOR RECOMBINED LEVEL 'IA()'
C                           NOTE: (ISA-1)/2 = QUANTUM NUMBER (S)
C  OUTPUT: (I*4)  ILA()   = QUANTUM NUMBER (L) FOR RECOMBINED LEVEL
C                           'IA()'
C  OUTPUT: (R*8)  XJA()   = QUANTUM NUMBER (J) FOR RECOMBINED LEVEL
C                           'IA()'
C                           NOTE: (2*XJA)+1 = STATISTICAL WEIGHT
C  OUTPUT: (R*8)  WA()    = ENERGY RELATIVE TO RECOMBINED LEVEL 1 (CM-1)
C                           FOR RECOMBINED LEVEL 'IA()'
C  OUTPUT: (L*4)  LRAUG   = .TRUE.  => RESOLVED AUGER DATA PRESENT
C                           .FALSE. => RESOLVED AUGER DATA NOT PRESENT
C  OUTPUT: (I*4)  NPF     = NUMBER OF FINAL PARENTS WITH RESOLVED AUGER DATA
C  OUTPUT: (I*4)  INDF()  = INDICES OF FINAL PARENTS WITH RESOLVED AUGER DATA
C  OUTPUT: (R*8)  RAUG(,) = RESOLVED AUGER RATES
C                           1ST.DIM: IONISING ION LEVEL INDEX
C                           2ND.DIM: FINAL PARENT INDEX
C  OUTPUT: (I*4)  IMETI() = INDEX OF METASTABLES IN IONISING ION LEVEL LIST
C  OUTPUT: (L*4)  LSYSM() = .TRUE.  => SPIN SYSTEMS LINKED TO METASTABLE
C                           .FALSE. => NO SPIN SYSTEMS LINKED TO METASTABLE
C  OUTPUT: (I*4)  NSYSM()  = NUMBER OF SPIN SYSTEMS LINKED TO METASTABLE
C  OUTPUT: (L*4)  LREPM() = .TRUE.  => REPR. LEVELS LINKED TO METASTABLE
C                           .FALSE. => NO REP. LEVELS LINKED TO METASTABLE
C  OUTPUT: (I*4)  NREPM()  = NUMBER OF REPR. LEVELS LINKED TO METASTABLE
C  OUTPUT: (L*4)  LIONIS()= .TRUE.  => STATE SELECTIVE IONIS. DATA PRESENT
C                           .FALSE. => STATE SELECTIVE IONIS. DATA NOT PRESENT
C                           1ST.DIM: INITIAL METASTABLE INDEX
C  OUTPUT: (L*4)  LRION(,)= .TRUE.  => DATA PRESENT FOR FINAL STATE
C                           .FALSE. => DATA NOT PRESENT FOR FINAL STATE
C                           1ST.DIM: IONISING ION METASTABLE INDEX
C                           2ND.DIM: FINAL PARENT INDEX
C  OUTPUT: (R*8)  RION(,,)= STATE SELECTIVE DIRECT IONISATION COEFFICIENTS
C                           1ST.DIM: IONISING ION METASTABLE INDEX
C                           2ND.DIM: FINAL PARENT INDEX
C                           3RD.DIM: ELECTRON TEMPERATURE INDEX
C  OUTPUT: (L*4)  LEXCIT()= .TRUE.  => STATE SELECTIVE EXCIT. DATA PRESENT
C                           .FALSE. => STATE SELECTIVE EXCIT. DATA NOT PRESENT
C                           1ST.DIM: INITIAL METASTABLE INDEX
C  OUTPUT: (L*4)  LREXC(,)= .TRUE.  => DATA PRESENT FOR FINAL STATE
C                           .FALSE. => DATA NOT PRESENT FOR FINAL STATE
C                           1ST.DIM: IONISING ION METASTABLE INDEX
C                           2ND.DIM: FINAL IONISING ION LEVEL INDEX
C  OUTPUT: (R*8)  REXC(,,)= STATE SELECTIVE DIRECT EXCITATION COEFFICIENTS
C                           1ST.DIM: IONISING ION METASTABLE INDEX
C                           2ND.DIM: FINAL IONISING ION LEVEL INDEX
C                           3RD.DIM: ELECTRON TEMPERATURE INDEX
C  OUTPUT: (I*4)  NPIS()  = INDEX OVER INTERMEDIATE PARENT & SPIN SYSTEM
```

```
C                             FOR BUNDLE-N DOUBLY EXCITED STATES
C                             1ST.DIM: IONISING ION METASTABLE INDEX
C  OUTPUT: (I*4)  IPRTI(,)= INTERMEDIATE PARENT INDEX IN PARENT LIST
C                             1ST.DIM: IONISING ION METASTABLE INDEX
C                             2ND.DIM: INTERMEDIATE PARENT & SPIN SYSTEM
C                                      INDEX
C  OUTPUT: (I*4)  ISYSI(,)= SPIN SYSTEM INDEX FOR INTERMEDIATE PARENT
C                             1ST.DIM: IONISING ION METASTABLE INDEX
C                             2ND.DIM: INTERMEDIATE PARENT & SPIN SYSTEM
C                                      INDEX
C  OUTPUT: (I*4)  ISPSYI(,)= SPIN OF SYSTEM FOR INTERMEDIATE PARENT
C                             1ST.DIM: IONISING ION METASTABLE INDEX
C                             2ND.DIM: INTERMEDIATE PARENT & SPIN SYSTEM
C                                      INDEX
C  OUTPUT: (I*4)  IMETF(,)= FINAL PARENT INDICES TO WHICH INTERM. PARENT/
C                             SPIN SYSTEM/REPR. N-SHELL AUGERS
C                             1ST.DIM: IONISING ION METASTABLE INDEX
C                             2ND.DIM: INTERMEDIATE PARENT & SPIN SYSTEM
C                                      INDEX
C                             3RD.DIM: FINAL IONISED ION METASTABLE INDEX
C  OUTPUT: (I*4)  NVALS(,)= NUMBER OF FINAL PARENT S TO WHICH INTERM.
C                             PARENT/SPIN SYSTEM/REPR. N-SHELL AUGERS
C                             1ST.DIM: IONISING ION METASTABLE INDEX
C                             2ND.DIM: INTERMEDIATE PARENT & SPIN SYSTEM
C                                      INDEX
C  OUTPUT: (I*4)  NREPI(,)= NUMBER OF REPR. N-SHELLS FOR  INTERM.
C                             PARENT/SPIN SYSTEM
C                             1ST.DIM: IONISING ION METASTABLE INDEX
C                             2ND.DIM: INTERMEDIATE PARENT & SPIN SYSTEM
C                                      INDEX
C  OUTPUT: (I*4)  IREP(,,)= REPR. N-SHELLS FOR INTERM. PARENT/SPIN SYSTEM/
C                             N-SHELL AUTOIONISING LEVELS
C                             1ST.DIM: REPRESENTATIVE N-SHELL INDEX
C                             2ND.DIM: IONISING ION METASTABLE INDEX
C                             3RD.DIM: INTERMEDIATE PARENT & SPIN SYSTEM
C                                      INDEX
C  OUTPUT: (R*8)  AUGN(,,,)=AUGER RATES FOR INTERM. PARENT/SPIN SYSTEM/
C                             REPRESENTATIVE. N-SHELL
C                             1ST.DIM: REPRESENTATIVE N-SHELL INDEX
C                             2ND.DIM: IONISING ION METASTABLE INDEX
C                             3RD.DIM: INTERMEDIATE PARENT & SPIN SYSTEM
C                                      INDEX
C  OUTPUT: (L*4)  LEXCN(,,)=.TRUE. => DATA PRESENT FOR REPR. N-SHELL
C                             .FALSE.=> DATA NOT PRESENT FOR REPR. N-SHELL
C                             1ST.DIM: REPRESENTATIVE N-SHELL INDEX
C                             2ND.DIM: IONISING ION METASTABLE INDEX
C                             3RD.DIM: INTERMEDIATE PARENT & SPIN SYSTEM
C                                      INDEX
C  OUTPUT: (R*8)  EXCN(,,,)=EXCITATION RATES TO AUTOIONISING PARENT/SPIN
C                             SYSTEM REPRESENTATIVE N-SHELL
C                             1ST.DIM: REPRESENTATIVE N-SHELL INDEX
C                             2ND.DIM: IONISING ION METASTABLE INDEX
C                             3RD.DIM: INTERMEDIATE PARENT & SPIN SYSTEM
C                                      INDEX
C                             4TH DIM: TEMPERATURE INDEX
C  OUTPUT: (I*4)  NTE     = NUMBER OF ELECTRON TEMPERATURES
C  OUTPUT: (R*8)  TEA()   = ELECTRON TEMPERATURES (K)
C
C          (I*4)  IND     = GENERAL INDEX
C          (I*4)  INDX    = GENERAL INDEX
C          (I*4)  INDX1   = GENERAL INDEX
```

```
C          (I*4)  INDX2   = GENERAL INDEX
C          (I*4)  II      = GENERAL INDEX
C          (I*4)  I       = GENERAL INDEX
C          (I*4)  IPI     = GENERAL INDEX FOR INTERM. PARENT
C          (I*4)  IPF     = GENERAL INDEX FOR FINAL PARENT
C          (I*4)  IPIS    = GENERAL INDEX FOR INTERM. PARENT/SPIN SYSTEM
C          (I*4)  IR      = GENERAL INDEX FOR REPRESENTATIVE N-SHELLS
C          (I*4)  IT      = GENERAL INDEX
C          (I*4)  J       = GENERAL INDEX
C          (I*4)  K       = GENERAL INDEX
C
C          (L*4)  LDATA   = GENERAL READ/DO NOT READ FLAG
C
C          (C*18) C18     = GENERAL CHARACTER STRING
C
C  ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4UNIT     ADAS      FETCH UNIT NUMBER FOR OUTPUT OF MESSAGES
C          I4EIZ0     ADAS      RETURNS NUCL. CHARGE FROM ELEMENT SYMBOL
C          R8FCTN     ADAS      CONVERTS FROM CHARACTER TO REAL VARIABLE
C          XXWORD     ADAS      EXTRACT POSITION OF NUMBER IN BUFFER
C
C  AUTHOR:  H. P. SUMMERS, UNIVERSITY OF STRATHCLYDE
C          JA8.08
C          TEL. 0141-553-4196
C
C  DATE:    21/03/98
C
C  UPDATE:
C
C VERSION: 1.1 DATE: 11-06-98
C MODIFIED: RICHARD MARTIN
C - PUT UNDER SCCS CONTROL.
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      CHARACTER*(*)       CSTRGA(NDLEV),            CSTRPA(NDPRT)
      CHARACTER*2         SEQSYM
      INTEGER             IA(NDLEV),   ILA(NDLEV),  ILEVM(NDLEV)
      INTEGER             ILPA(NDPRT), IMETF(NDMET,NDPRTI,NDPRT)
      INTEGER             IMETI(NDMET),             INDF(NDPRT)
      INTEGER             IPA(NDPRT),  IPRFM(NDPRT)
      INTEGER             IPRI(NDPRTI),             IPRTI(NDMET,NDPRTI)
      INTEGER             IREP(NDREP,NDMET,NDPRTI), ISA(NDLEV)
      INTEGER             ISPA(NDPRT), ISPSYI(NDMET,NDPRTI)
      INTEGER             ISYSI(NDMET,NDPRTI),      IUNIT,       IZ
      INTEGER             IZ0,         IZ1,         NDAUG,       NDLEV
      INTEGER             NDMET,       NDPRT,       NDPRTI,      NDREP
      INTEGER             NDT,         NLEV,        NLEVM,       NPF
      INTEGER             NPIS(NDMET), NPRF,        NPRFM,       NPRI
      INTEGER             NREPI(NDMET,NDPRTI),      NREPM(NDMET)
      INTEGER             NSYSM(NDMET),             NTE
      INTEGER             NVALS(NDMET,NDPRTI)
      LOGICAL             LEXCIT(NDMET)
      LOGICAL             LEXCN(NDREP,NDMET,NDPRTI)
      LOGICAL             LIONIS(NDMET),            LRAUG
      LOGICAL             LREPM(NDMET),             LREXC(NDMET,NDLEV)
      LOGICAL             LRION(NDMET,NDPRT),       LSYSM(NDMET)
      REAL*8              AUGN(NDREP,NDMET,NDPRTI,NDPRT),       BWNF
      REAL*8              BWNI,        EXCN(NDREP,NDMET,NDPRTI,NDT)
```

```
REAL*8              RAUG(NDLEV,NDPRT)
REAL*8              REXC(NDMET,NDLEV,NDT)
REAL*8              RION(NDMET,NDPRT,NDT),     TEA(NDT)
REAL*8              WA(NDLEV),   WPA(NDPRT),   XJA(NDLEV)
REAL*8              XJPA(NDPRT)
```

5.80 ddspln: Subroutine ddspln from library adas4xx

```
      SUBROUTINE DDSPLN( NTDIM  , NDTIN  ,
     &                   ITA    , ITVAL  ,
     &                   TFILE  , TEVA   ,
     &                   QDRIN  , QDROUT ,
     &                   LTRNG
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTFILEE: DDSPLN *********************
C
C  PURPOSE:
C         1) PERFORMS CUBIC SPLINE ON LOG(TEMPERATURE) VERSUS LOG(QDRIN)
C            INPUT DATA. ('TFILE' VERSUS 'QDRIN' , ITA DATA PAIRS)
C
C         2) INTERPOLATES 'ITVAL'  QDRIN VALUES USING ABOVE SPLINES AT
C            TEMPERATURES READ IN FROM ISPF PANELS FOR TABULAR OUTPUT.
C            (ANY TEMPERATURE VALUES WHICH REQUIRED EXTRAPOLATION TO
C             TAKE PLACE ARE SET TO ZERO).
C
C         3) INTERPOLATES 'ITVAL' QDRIN VALUES USING ABOVE SPLINES AT
C            TEMPERATURES EQUI-DISTANCE ON RANGE OF LOG(TEMPERATURES)
C            STORED IN INPUT 'TFILE' ARRAY.
C
C  CALLING PROGRAM: ADAS413
C
C
C  SUBROUTFILEE:
C
C  INPUT : (I*4)  NTDIM   = MAX. NO. OF TEMPERATURES ALLOWED IN
C                           DATA SET
C  INPUT : (I*4)  NDTIN   = MAX. NO. OF USER TEMPERATURES ALLOWED
C  INPUT : (I*4)  ITA     = INPUT DATA FILE: NUMBER OF DR/TEMPERATURE
C                           PAIRS READ FOR THE TRANSITION BEING ASSESSED
C  INPUT : (I*4)  ITVAL   = NUMBER OF  SPLINE  INTERPOLATED  QDRIN/TEMP.
C                           REQUIRED FOR GRAPHICAL DISPLAY.
C
C  INPUT : (I*4)  TFILE() = INPUT DATA FILE: TEMPERATURES
C  INPUT : (I*4)  TEVA()  = ISPF PANEL ENTERED TEMPERATURES
C
C  INPUT : (R*8)  QDRIN() = INPUT DATA FILE: SELECTED TRANSITION -
C                           QDRIN VALUES AT 'TFILE()'.
C  OUTPUT: (I*4)  QDROUT()= SPLINE INTERPOLATED QDRIN VALUES AT 'TEVA()'
C                           (EXTRAPOLATED VALUES = 0.0).
C
C  OUTPUT: (L*4)  LTRNG() = .TRUE.  => OUTPUT   SPLINE    VALUE    WAS
C                                      INTERPOLATED FOR 'DLOG(TEVA()'.
C                           .FALSE. => OUTPUT   SPLINE    VALUE    WAS
C                                      EXTRAPOLATED FOR 'DLOG(TEVA()'.
C                                      (NOTE: 'YOUT()=0' AS 'IOPT < 0').
C
C          (I*4)  NIN     = PARAMETER = MAX. NO. OF  INPUT  TEMP/QDRIN
C                                      PAIRS MUST BE >= 'ITA'
C          (I*4)  NOUT    = PARAMETER = MAX. NO. OF 'OUTPUT TEMP/QDRIN
C                                      PAIRS MUST BE >= 'ITVAL' & 'ITVAL'
C
C          (I*4)  IARR    = ARRAY SUBSCRIPT USED FOR TEMP/QDRIN PAIRS
C          (I*4)  IOPT    = DEFINES THE BOUNDARY DERIVATIVES FOR THE
C                           SPLINE ROUTFILEE 'XXSPLE', SEE 'XXSPLE'.
C                           (VALID VALUES = <0, 0, 1, 2, 3, 4)
```

```
C
C          (L*4)  LSETX   = .TRUE.  => SET UP SPLINE PARAMETERS RELATFILEG
C                                     TO 'XIN' AXIS.
C                           .FALSE. => DO NOT SET UP SPLINE PARAMETERS
C                                     RELATFILEG TO 'XIN' AXIS.
C                                     (I.E. THEY WERE SET IN A PREVIOUS
C                                           CALL )
C                           (VALUE SET TO .FALSE. BY 'XXSPLE')
C
C          (R*8)  XIN()   = LOG( 'TFILE()' )
C          (R*8)  YIN()   = LOG( 'QDRIN()' )
C          (R*8)  XOUT()  = LOG(TEMPERATURES AT WHICH SPLINES REQUIRED)
C          (R*8)  YOUT()  = LOG(OUTPUT SPLINE INTERPOLATED QDRIN VALUES)
C          (R*8)  DF()    = SPLINE INTERPOLATED DERIVATIVES
C
C
C NOTE:
C
C ROUTFILEES:
C          ROUTFILEE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          XXSPLE     ADAS      SPLINE SUBROUTFILEE (EXTENDED DIAGNOSTICS)
C          R8FUN1     ADAS      REAL*8 FUNCTION: ( X -> X )
C
C AUTHOR:  H. P. Summers, University of Strathclyde
C          JA8.08
C          Tel. 0141-553-4196
C
C
C DATE:    30/01/98
C
C UPDATE:
C
C VERSION: 1.1 DATE: 11-06-98
C MODIFIED: RICHARD MARTIN
C - PUT UNDER SCCS CONTROL.
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             ITA,         ITVAL,       NDTIN,       NTDIM
      LOGICAL             LTRNG(NDTIN)
      REAL*8              QDRIN(NTDIM),             QDROUT(NDTIN)
      REAL*8              TEVA(NDTIN), TFILE(NTDIM)
```

5.81 ddtitl: Subroutine ddtitl from library adas4xx

```
      SUBROUTINE DDTITL( IOPT     ,
     &             IP_RES   , IL_RES ,
     &   DSFULL   ,
     &                   TITLX
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: DDTITL *********************
C
C  PURPOSE:  TO CREATE THE DESCRIPTIVE TITLE FOR SELECTED DATA-BLOCK.
C
C  CALLING PROGRAM: ADAS413
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  IOPT       = Type of plot
C  INPUT : (I*4)  IL_RES     = resolved DR parent
C  INPUT : (I*4)  IL_RES     = resolved DR level
C  I
C  INPUT : (C*80) DSFULL   = FULL INPUT DATA SET NAME
C
C
C  INPUT : (C*9)  CDONOR   = SELECTED DATA-BLOCK: DONOR IDENTITY
C
C  INPUT : (C*9)  CRECVR   = SELECTED DATA-BLOCK: RECEIVER IDENTITY
C
C  INPUT : (C*10) CFSTAT  = SELECTED DATA-BLOCK: FINAL STATE SPEC.
C
C  OUTPUT: (C*120) TITLX    = SELECTED DATA-BLOCK: DESCRIPTIVE TITLE
C
C          (C*2)  C2       = GENERAL USE 2 BYTE  CHARACTER  STRING
C
C
C AUTHOR   : H. P. Summers, University of Strathclyde
C            JA8.08
C            Tel. 0141-553-4196
C
C DATE     : 30/01/98
C
C UPDATE:
C
C VERSION: 1.1 DATE: 11-06-98
C MODIFIED: RICHARD MARTIN
C - PUT UNDER SCCS CONTROL.
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      CHARACTER*120       DSFULL,      TITLX
      INTEGER             IL_RES,      IOPT,        IP_RES
```

5.82 dhdata: Subroutine dhdata from library adas4xx

```
       SUBROUTINE DHDATA( YEAR   , YEARDF , TITLF  , IFAIL
     &                  , IZ0    , IZ1    , ICLASS , ITMAX  , IEVCUT
     &                  , ITDIMD , ITMAXD , IDMAXD , IZMAXD
     &                  , DTEV   , DDENS
     &                  , DTEVD  , DDENSD , DRCOFD , ZDATA
     &                  , DRCOFI
     &                  )
C
C-----------------------------------------------------------------------
C  ****************** FORTRAN77 SUBROUTINE: DHDATA *********************
C
C PURPOSE : TO EXTRACT 'SANC0' COLLISIONAL DIELECTRONIC DATA
C
C CALLING PROGRAM: IONBAL (ADAS412)
C
C NOTE    : THE SOURCE DATA IS CONTAINED AS SEQUENTIAL DATASETS
C           AS FOLLOWS:
C
C                  (1) JETSHP.ACD<YR>#<IEL).DATA
C                  (2) JETSHP.SCD<YR>#<IEL>.DATA
C                  (3) JETSHP.CCD<YR>#<IEL>.DATA
C                  (4) JETSHP.PRB<YR>#<IEL>.EV<CUT>.DATA
C                  (5) JETSHP.PLT<YR>#<IEL>.EV<CUT>.DATA
C                  (6) JETSHP.PRC<YR>#<IEL>.EV<CUT>.DATA
C                  (7) JETSHP.PLS<YR>#<IEL>.DATA
C
C       WHERE, <YR>  = TWO INTEGERS  FOR THE YEAR SELECTED
C              <IEL> = ELEMENT NAME
C              <CUT> = ENERGY CUT-OFF (EV)
C
C              IF <CUT> = 0 THEN .EV<CUT> IS DELETED FROM ABOVE FILES.
C
C#
C# mar20-95 Alessandro Lanzafame
C#          conversion to Unix
C#
C#                  (1) /ADAS/adas/adf11/acd<YR>/acd<YR>_<IEL>.dat
C#                  (2) /ADAS/adas/adf11/scd<YR>/scd<YR>_<IEL>.dat
C#                  (3) /ADAS/adas/adf11/ccd<YR>/ccd<YR>_<IEL>.dat
C#                  (4) /ADAS/adas/adf11/prb<YR>/prb<YR>_<IEL>.dat
C#                  (5) /ADAS/adas/adf11/plt<YR>/plt<YR>_<IEL>.dat
C#                  (6) /ADAS/adas/adf11/prc<YR>/prc<YR>_<IEL>.dat
C#                  (7) /ADAS/adas/adf11/pls<YR>/pls<YR>_<IEL>.dat
C#
C INPUT  : (C*2)  YEAR      = YEAR OF DATA
C          (C*2)  YEARDF    = DEFAULT YEAR OF DATA IF REQUESTED YEAR
C                             DOES NOT EXIST
C          (I*4)  IZ0       = NUCLEAR CHARGE
C          (I*4)  IZ1       = MINIMUM ION CHARGE + 1
C          (I*4)  ICLASS    = CLASS OF DATA (1 - 6)
C          (I*4)  ITMAX     = NUMBER OF ( DTEV() , DDENS() ) PAIRS
C          (I*4)  IEVCUT    = ENERGY CUT-OFF (EV)
C          (R*8)  DTEV()    = DLOG10(ELECTRON TEMPERATURES (EV))
C          (R*8)  DDENS()   = DLOG10(ELECTRON DENSITIES (CM-3))
C
C OUTPUT : (C*80) TITLF     = INFORMATION STRING
C          (I*4)  ITDIMD    = MAXIMUM NUMBER OF DATA TEMP & DENS
C          (I*4)  ITMAXD    = NUMBER OF DATA DTEVD()
C          (I*4)  IDMAXD    = NUMBER OF DATA DDENS()
```

```
C          (I*4)  IZMAXD    = NUMBER OF DATA ZDATA()
C          (I*4)  ITDIMD    = MAXIMUM NUMBER OF DATA TEMP & DENS
C          (I*4)  ZDATA()   = Z1 CHARGES IN DATASET
C          (I*4)  IFAIL     = 0    IF ROUTINE SUCCESSFUL
C                           = 1    IF ROUTINE OPEN STATEMENT FAILED
C          (R*8)  DTEVD()   = DLOG10(DATA ELECTRON TEMPERATURES (EV))
C          (R*8)  DDENSD()  = DLOG10(DATA ELECTRON DENSITIES (CM-3))
C          (R*8)  DRCOFD()  = DLOG10(DATA RATE COEFFICIENTS (CM-3/S))
C          (R*8)  DRCOFI()  = INTERPOLATION OF DRCOFD(,,) FOR
C                             DTEV() & DDENS()
C
C PROGRAM: (C*2)  SEQUA()   = ION NAMES FOR A PARTICULAR IZ0
C          (C*36) DSNAME    = FILE NAME ( SEE ABOVE TYPES )
C          (C*80) STRING    = GENERAL VARIABLE
C          (C*80) BLANK     = BLANK STRING
C          (C*2)  YEARSV    = LAST YEAR USED IN THIS ROUTINE
C          (I*4)  IREAD     = INPUT STREAM FOR OPEN STATEMENT
C          (I*4)  IZ0SV     = LAST IZ0 USED IN THIS ROUTINE
C          (I*4)  ICLSV     = LAST ICLASS USED IN THIS ROUTINE
C          (I*4)  INDXZ1    = LOCATION OF IZ1 IN ZDATA()
C          (I*4)  LCK       = MUST BE GREATER THAN 'ITMAXD' & 'IDMAXD'
C                             & 'ITMAX' - ARRAY SIZE FOR SPLINE CALCS.
C          (R*8)  A()       = GENERAL ARRAY
C          (R*8)  DRCOF0(,) = INTERPOLATION OF DRCOFD(,,) W.R.T DTEV()
C          (L*8)  LEXIST    = TRUE --- FILE TO OPEN EXISTS ELSE NOT
C
C PE BRIDEN = ADDED VARIABLES (14/01/91)
C
C          (I*4)  L1      = PARAMETER = 1
C          (I*4)  IOPT    = DEFINES THE BOUNDARY DERIVATIVES FOR THE
C                             SPLINE ROUTINE 'XXSPLN', SEE 'XXSPLN'.
C
C          (L*4)  LSETX   = .TRUE.  => SET UP SPLINE PARAMETERS RELATING
C                                      TO X-AXIS.
C                           .FALSE. => DO NOT SET UP SPLINE PARAMETERS
C                                      RELATING TO X-AXIS.
C                                      (I.E. THEY WERE SET IN A PREVIOUS
C                                            CALL )
C                           (VALUE SET TO .FALSE. BY 'XXSPLN')
C
C
C          (R*8)  DY()    = SPLINE INTERPOLATED DERIVATIVES
C
C          (R*8 ADAS FUNCTION - 'R8FUN1' ( X -> X) )
C
C AUTHOR : JAMES SPENCE (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/80
C          JET  EXT. 4866
C
C DATE   : 22/02/90
C
C DATE   : 21/08/90 PE BRIDEN - REVISION: SEQUA(43) CHANGED ('TE'->'TC')
C
C DATE   : 08/10/90 PE BRIDEN - REVISION: RENAMED SUBROUTINE
C
C DATE   : 12/11/90 PE BRIDEN - CORRECTION: MOVE THE SETTING OF 'INDXZ1'
C                                           TO AFTER THE  '20 CONTINUE'
C                                           STATEMENT.   ALSO SAVE  THE
C                                           VALUE OF 'IZ1MIN'.
C
C DATE   : 14/01/91 PE BRIDEN - ADAS91:     CALLS TO NAG SPLINE ROUTINES
```

```
C                                          'E01BAF' & 'E02BBF' REPLACED
C                                          BY  CALLS   TO  ADAS  SPLINE
C                                          ROUTINE 'XXSPLN'.
C
C DATE   : 25/06/91 PE BRIDEN - CORRECTION: CHANGED FOLLOWING DIMENSION:
C                                           'DIMENSION DRCOFI(ITDIMD)'
C                                          TO
C                                           'DIMENSION DRCOFI(ITMAX)'
C
C
C DATE   : 25/06/91 HP SUMMERS - REVISION: RENAMED FROM D2DATA TO DHDATA
C                                          SET IOPT=4 FOR IONBAL
C DATE   : 27/04/92 PE BRIDEN  - ADDED DEFAULT YEAR FOR DATA IF
C                                REQUESTED YEAR DOES NOT EXIST.
C                                (ADDED 'YEARDF'), INTRODUCED IFAIL=-1
C                                IF DEFAULT YEAR WAS USED AND NOT THE
C                                REQUESTED YEAR.
C
C
C DATE   : 20/03/95 AC LANZAFAME - CONVERSION TO UNIX
C                                  CHANGED OPEN STATEMENT AND FILE
C                                  NAMES
C                                  ADIR ADIR1 ADDED
C                                  DSNAME FROM C*30 TO C*40
C
C  ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          XXSPLN     ADAS      CUBIC SPLINE INTERPOLATION/EXTRAPOLATION
C      R8FUN1     ADAS      REAL*8 FUNCTION: ( X -> X )
C
C VERSION 1.1 DATE: 29-10-97
C RICHARD MARTIN
C ADAS412 - PUT UNDER SCCS CONTROL.
C
C VERSION: 1.2 DATE: 01-12-97
C RICAHRD MARTIN
C REMOVED DUPLICATE VARIABLE DECLARATION.
C
C VERSION: 1.3 DATE: 23-11-98
C RICHARD MARTIN & DAVID BROOKS
C REMOVED 'DSNAME(12:13) = YEARDF' STATEMENT (FOR IBM).
C
C-----------------------------------------------------------------------
C
      CHARACTER*80        TITLF
      CHARACTER*2         YEAR,        YEARDF
      INTEGER             ICLASS,      IDMAXD,      IEVCUT,      IFAIL
      INTEGER             ITDIMD,      ITMAX,       ITMAXD,      IZ0
      INTEGER             IZ1,         IZMAXD,      ZDATA(ITDIMD)
      REAL*8              DDENS(ITMAX),             DDENSD(ITDIMD)
      REAL*8              DRCOFD(ITDIMD,ITDIMD,ITDIMD)
      REAL*8              DRCOFI(ITMAX),            DTEV(ITMAX)
      REAL*8              DTEVD(ITDIMD)
```

5.83 dxcomp: Subroutine dxcomp from library adas4xx

```
C UNIX-IDL PORT - SCCS INFO: MODULE @(#)dxcomp.for 1.1 DATE 08/20/96

      SUBROUTINE DXCOMP( NOCCUM , NOCCUP , LMATCH , IORBIT )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN 77 SUBROUTINE: DXCOMP *****************
C
C  PURPOSE: COMPARES PARENT AND RECOMBINE ION METASTABLE CONFIGURATION
C           EXPANDED ORBITAL VECTORS.  RETURNS .TRUE. IF ONE DIFFERENCE
C           AND GIVES DIFFERING ORBITAL.
C
C  CALLING PROGRAM: D7LINK
C
C  SUBROUTINE:
C
C  INPUT  :  (I*4)   NOCCUM() = OCCUPANCY FOR EACH DECIMAL ORBITAL
C                               INDEX 1-15 OF RECOMBINED ION METASTABLE
C  INPUT  :  (I*4)   NOCCUP() = OCCUPANCY FOR EACH DECIMAL ORBITAL
C                               INDEX 1-15 OF PARENT METASTABLE
C
C  OUTPUT :  (L*4)   LMATCH   = .TRUE.  => ONE ORBITAL DIFFERENCE
C                               .FALSE. => 0 OR >1 MISMATCH
C  OUTPUT :  (L*4)   IORBIT   = DECIMAL ORBITAL INDEX FOR MISMATCH
C
C            (I*4)   I      = GENERAL INDEX
C
C  ROUTINES:  NONE
C
C
C  AUTHOR:  H. P. SUMMERS, UNIVERSITY OF STRATHCLYDE
C           JA8.08
C           TEL. 0141-553-4196
C
C  DATE:    05/06/96
C
C UNIX-IDL PORT:
C    WILLIAM OSBORN, TESSELLA SUPPORT SERVICES PLC.
C
C DATE:    20TH AUGUST 1996
C
C VERSION: 1.1 DATE: 20-08-96
C MODIFIED: WILLIAM OSBORN
C     - FIRST VERSION
C
C VERSION: 1.2 DATE: 22-11-2003
C MODIFIED: Martin O'Mullane
C     - Extend NOCCUM and NOCCUP to 61 (from 15).
C
C-----------------------------------------------------------------------
      INTEGER             IORBIT,      NOCCUM(61),  NOCCUP(61)
      LOGICAL             LMATCH
```

5.84 dxexcf: Subroutine dxexcf from library adas4xx

```
      SUBROUTINE DXEXCF( CONFG , NOCCUP ,LTYPE)
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN 77 SUBROUTINE: DXEXCF *******************
C
C  PURPOSE: RETURNS VECTOR OF OCCUPANCIES FOR STANDARD SHELL INDICES
C           1-15 FROM AN EISSNER HEXADECIMAL CHARACTER CONFIGURATION
C           SPECIFICATION
C
C  CALLING PROGRAM: VARIOUS
C
C  SUBROUTINE:
C
C  INPUT  :  (C*18)  CONFIG   = EISSNER CONFIGURATION
C  OUTPUT :  (I*4)   NOCCUP() = OCCUPANCY FOR EACH DECIMAL ORBITAL
C                               INDEX 1-15.
C  OUTPUT :  (L*4)   LTYPE    = .TRUE.  => CONFIG. EISSNER FORM
C                               .FALSE. => CONFIG. NOT EISSNER FORM
C
C            (I*4)   I      = GENERAL INDEX
C            (I*4)   IFIRST = GENERAL STRING POSITION INDEX
C            (I*4)   ILAST  = GENERAL STRING POSITION INDEX
C            (C*1)   CHRA() = EISSNER HEXADECIMAL ORBITAL LIST
C            (L*4)   LSTAN  = .TRUE.  => CONFIG. STANDARD FORM
C                             .FALSE. => CONFIG. NOT STANDARD FORM
C            (I*4)   NVLCE  = VALENCE SHELL
C
C  ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4EISS     ADAS      EXPAND EISSNER CONFIG. INTO SHELL OCCUP.
C          I4UNIT     ADAS      FETCH UNIT NUMBER FOR OUTPUT OF MESSAGES
C          XXDTES     ADAS      DETECT CONFIGURATION FORM
C          XXSLEN     ADAS      FIND NON-BLANK LENGTH OF A STRING
C
C  AUTHOR:  H. P. SUMMERS, UNIVERSITY OF STRATHCLYDE
C           JA8.08
C           TEL. 0141-553-4196
C
C  DATE:    04/06/96
C
C UNIX-IDL PORT:
C    WILLIAM OSBORN, TESSELLA SUPPORT SERVICES PLC.
C
C DATE:    20TH AUGUST 1996
C
C VERSION: 1.1 DATE: 20-08-96
C MODIFIED: WILLIAM OSBORN
C     - FIRST VERSION
C
C VERSION: 1.2
C MODIFIED: H. P. SUMMERS DATE: 24-06-97
C       - INSERTED TRAP FOR BARE NUCLEUS CONFIG.
C
C VERSION: 1.3
C MODIFIED: R. MARTIN           DATE: 30-06-97
C     - ADDED SCCS KEYWORDS OMITTED IT PREVIOUS VERSION.
C
C VERSION: 1.4 DATE: 22-11-2003
```

```
C MODIFIED: Martin O'Mullane
C     - Pass configurations through ceprep before acting on them.
C
C-----------------------------------------------------------------------
      CHARACTER*18        CONFG
      INTEGER             NOCCUP(61)
      LOGICAL             LTYPE
```

5.85 dxmadd: Subroutine dxmadd from library adas4xx

```
       SUBROUTINE DXMADD( NDMET,
     &                    A    , NRA  , NCA  , B   , NRB  , NCB  ,
     &                    C    , NRC  , NCC  ,
     &                    NX   , NY
     &                  )
C
C-----------------------------------------------------------------------------
C                                                                        ********
C  ****************** FORTRAN 77 SUBROUTINE: DXMADD ***************************
C
C  PURPOSE: CALCULATES THE SUM OF TWO RECTANGULAR MATRICES WITH
C           ARBITRARY INTEGER MULTIPLIERS
C
C  CALLING PROGRAM: D5MFSP
C
C  SUBROUTINE:
C
C  INPUT :(I*4) NDMET         = MAXIMUM DIMENSION OF MATRICES
C  INPUT :(R*8) A(,)          = FIRST MATRIX
C  INPUT :(I*4) NRA           = NUMBER OF ROWS IN MATRIX A
C  INPUT :(I*4) NCA           = NUMBER OF COLUMNS IN MATRIX A
C  INPUT :(R*8) B(,)          = SECOND MATRIX
C  INPUT :(I*4) NRB           = NUMBER OF ROWS IN MATRIX B
C  INPUT :(I*4) NCB           = NUMBER OF COLUMNS IN MATRIX B
C  INPUT :(I*4) NX            = MULTIPLIER OF FIRST MATRIX
C  INPUT :(I*4) NY            = MULTIPLIER OF SECOND MATRIX
C  INPUT :(I*4) NRC           = NUMBER OF ROWS IN MATRIX C
C  INPUT :(I*4) NCC           = NUMBER OF COLUMNS IN MATRIX C
C
C  OUTPUT:(R*8) C(,)          = RESULTANT MATRIX C
C
C
C  ROUTINES: NONE
C
C
C  AUTHOR:  D. BROOKS, H. P. SUMMERS, JET
C           K1/1/57
C           JET EXT. 4941
C
C  DATE:    02/06/94
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 08-11-95
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST RELEASE
C
C-----------------------------------------------------------------------------
      INTEGER             NCA,         NCB,         NCC,         NDMET
      INTEGER             NRA,         NRB,         NRC,         NX
      INTEGER             NY
      REAL*8              A(NDMET,NDMET),           B(NDMET,NDMET)
      REAL*8              C(NDMET,NDMET)
```

5.86 dxmmul: Subroutine dxmmul from library adas4xx

```
       SUBROUTINE DXMMUL( NDMET ,
     &                    A     , NRA  , NCA  , B    , NRB  , NCB  ,
     &                    C
     &                  )
C
C-----------------------------------------------------------------------
C                                                                       ********
C  ****************** FORTRAN 77 SUBROUTINE: DXMMUL ***************************
C
C  PURPOSE: CALCULATES THE PRODUCT OF TWO RECTANGULAR MATRICES
C
C  CALLING PROGRAM: D5MFSP
C
C  SUBROUTINE:
C
C  INPUT :(I*4) NDMET         = MAXIMUM DIMENSION OF MATRICES
C  INPUT :(R*8) A(,)          = FIRST MATRIX
C  INPUT :(I*4) NRA           = NUMBER OF ROWS IN MATRIX A
C  INPUT :(I*4) NCA           = NUMBER OF COLUMNS IN MATRIX A
C  INPUT :(R*8) B(,)          = SECOND MATRIX
C  INPUT :(I*4) NRB           = NUMBER OF ROWS IN MATRIX B
C  INPUT :(I*4) NCB           = NUMBER OF COLUMNS IN MATRIX B
C
C  OUTPUT:(R*8) C(,)          = RESULTANT MATRIX C
C
C         (I*4) L             = GENERAL INDEX
C         (I*4) M             = GENERAL INDEX
C         (I*4) N             = GENERAL INDEX
C
C
C  ROUTINES: NONE
C
C
C  AUTHOR:  D. BROOKS, H. P. SUMMERS, JET
C           K1/1/57
C           JET EXT. 4941
C
C  DATE:    02/06/94
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 08-11-95
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST RELEASE
C
C-----------------------------------------------------------------------
      INTEGER             NCA,         NCB,         NDMET,       NRA
      INTEGER             NRB
      REAL*8              A(NDMET,NDMET),           B(NDMET,NDMET)
      REAL*8              C(NDMET,NDMET)
```

5.87 dxrdnm: Subroutine dxrdnm from library adas4xx

```
C
       SUBROUTINE DXRDNM( DSNINC , LPART  , IFAIL  ,
     &                    IZ0    , NPART  , IPRTD  , IGRDD , ICLASS ,
     &                    IZ1    , ITMAX  ,
     &                    ISDIMD , IZDIMD , ITDIMD ,
     &                    ISMAXD , IZMAXD , ITMAXD , IDMAXD , NPARTR,
     &                    DTEV   , DDENS  ,
     &                    DTEVD  , DDENSD , DRCOFD , ZDATA  ,
     &                    DRCOFI
     &                  )
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: DXRDNM *********************
C
C PURPOSE : TO EXTRACT COLLISIONAL DIELECTRONIC DATA  FROM
C           EITHER PARTIAL (METASTABLE/PARENT RESOLVED) OR STANDARD
C           (UNRESOLVED) ISONUCLEAR MASTER FILES
C
C NOTE    : THE SOURCE DATA IS CONTAINED AS SEQUENTIAL DATASETS
C           WITH THE FOLLOWING NAMING CONVENTIONS:
C
C                   (1) JETSHP.ACD<YR>#<EL).<CODE>DATA
C                   (2) JETSHP.SCD<YR>#<EL>.<CODE>DATA
C                   (3) JETSHP.CCD<YR>#<EL>.<CODE>DATA
C                   (4) JETSHP.PRB<YR>#<EL>.<FILT>.<CODE>DATA
C                   (5) JETSHP.PRC<YR>#<EL>.<FILT>.<CODE>DATA
C                   (6) JETSHP.QCD<YR>#<EL>.<CODE>DATA
C                   (7) JETSHP.XCD<YR>#<EL>.<CODE>DATA
C                   (8) JETSHP.PLT<YR>#<EL>.<CODE>DATA
C                   (9) JETSHP.PLS<YR>#<EL>.<CODE>DATA
C
C       WHERE, <YR>   = TWO DIGIT YEAR NUMBER
C              <EL>   = ONE OR TWO CHARACTER ELEMENT SYMBOL
C              <CODE> = R       => PARTIAL DATA
C                       U       => PARTIAL DATA
C                       OMITTED => STANDARD DATA
C              <FILT> = SIX CHARACTER POWER FILTER CODE
C
C       AND DATA OF CLASSES 6 AND 7 DO NOT EXIST FOR THE PARTIAL CASE.
C
C
C INPUT  : (C*120) DSNINC   = ISONUCLEAR MASTER FILE NAME - VERIFIED
C                             AND READY FOR DYNAMIC ALLOCATION.
C INPUT  : (L*4)  LPART     = .TRUE.  => PARTIAL (RESOLVED) MASTER DATA
C                            . FALSE. => UNSRESOLVED MASTER DATA
C INPUT  : (I*4)  IZ0       = NUCLEAR CHARGE
C INPUT  : (I*4)  NPART()   = METASTABLE PARTITION.  I.E. NUMBER OF
C                             METASTABLES FROM CHARGE STATE IZ1MIN-1 TO
C                             IZ1MAX ON INPUT
C INPUT  : (I*4)  IPRTD     = REQUIRED PARENT INDEX
C INPUT  : (I*4)  IGRDD     = REQUIRED GROUND INDEX
C INPUT  : (I*4)  ICLASS    = CLASS OF DATA (1 - 9 )
C INPUT  : (I*4)  IZ1       = REQUIRED ION CHARGE + 1
C INPUT  : (I*4)  ITMAX     = NUMBER OF ( DTEV() , DDENS() ) PAIRS
C INPUT  : (I*4)  ISDIMD    = MAXIMUM NUMBER OF (CHARGE, PARENT, GROUND)
C                             BLOCKS IN ISONUCLEAR MASTER FILES
C INPUT  : (I*4)  IZDIMD    = MAXIMUM NUMBER OF CHARGE STATES
C                             IN ISONUCLEAR MASTER FILES
```

```
C INPUT  : (I*4)  ITDIMD    = MAXIMUM NUMBER OF TEMP OR DENS VALUES IN
C                             ISOELECTRONIC MASTER FILES
C INPUT  : (R*8)  DTEV()    = DLOG10(ELECTRON TEMPERATURES (EV))
C INPUT  : (R*8)  DDENS()   = DLOG10(ELECTRON DENSITIES (CM-3))
C
C OUTPUT : (I*4)  IFAIL     = 0    IF ROUTINE SUCCESSFUL - DATA FOR THE
C                                  REQUESTED YEAR USED.
C                           = 1    IF ROUTINE OPEN STATEMENT FAILED
C                           = 2    IF FILE EXISTS BUT REQUIRED DATA
C                                  BLOCK DOES NOT
C OUTPUT : (I*4)  ISMAXD    = NUMBER OF (CHARGE, PARENT, METASTABLE)
C                             BLOCKS IN SELECTED MASTER FILE
C OUTPUT : (I*4)  IZMAXD    = NUMBER OF ZDATA() VALUES IN SELECTED
C                             MASTER FILE
C OUTPUT : (I*4)  ITMAXD    = NUMBER OF DTEVD() VALUES IN SELECTED
C                             MASTER FILE
C OUTPUT : (I*4)  IDMAXD    = NUMBER OF DDENSD() VALUES IN SELECTED
C                             MASTER FILE
C OUTPUT : (I*4)  NPARTR()  = METASTABLE PARTITION.  I.E. NUMBER OF
C                             METASTABLES FROM CHARGE STATE IZ1MIN-1 TO
C                             IZ1MAX FOUND IN MASTER FILE
C OUTPUT : (R*8)  DTEVD()   = DLOG10(DATA ELECTRON TEMPERATURES (EV))
C                             IN SELECTED MASTER FILE
C OUTPUT : (R*8)  DDENSD()  = DLOG10(DATA ELECTRON DENSITIES (CM-3))
C                             IN SELECTED MASTER FILE
C OUTPUT : (R*8)  DRCOFD(,,)= DLOG10(DATA RATE COEFFICIENTS (CM-3/S))
C                             IN SELECTED MASTER FILE
C                             1ST DIM: (CHARGE,META,GRD) BLOCK INDEX
C                             2ND DIM: TEMPERATURE INDEX
C                             3RD DIM: DENSITY INDEX
C OUTPUT : (R*8)  ZDATA()   = CHARGE + 1 FOR IONS IN SELECTED MASTER
C                             FILE
C                             1ST DIM: (CHARGE,META,GRD) BLOCK INDEX
C OUTPUT : (R*8)  DRCOFI()  = INTERPOLATION OF DRCOFD(,,) FOR
C                             DTEV() & DDENS()
C
C PROGRAM: (C*80) DSNOLD    = FILE NAME USED IN PREVIOUS CALL
C          (C*80) CLINE     = GENERAL CHARACTER VARIABLE
C          (C*80) CTERM     = TERMINATOR LINE - '-' FILLED VARIABLE
C          (C*4)) CPATRN()  = PATTERN USED TO DETECT DATA CLASS
C          (I*4)  IZ0D      = NUCLEAR CHARGE READ FROM MASTER FILE
C          (I*4)  IZ1MIN    = MINIMUM CHARGE+1 READ FROM MASTER FILE
C          (I*4)  IZ1MAX    = MAXIMUM CHARGE+1 READ FROM MASTER FILE
C          (I*4)  IABT      = ABORT CODE
C          (I*4)  INDSEL    = LOCATION OF (CHARGE,PRNT,GRND)
C                             DATA BLOCK IN FILE
C          (I*4)  IZDAT     = CURRENT DATA BLOCK ION CHARGE +1
C          (I*4)  ISEL      = GENERAL INDEX
C          (I*4)  I         = GENERAL INDEX
C          (I*4)  IT        = GENERAL INDEX
C          (I*4)  ID        = GENERAL INDEX
C          (I*4)  IZCHK     = INDEX TO VERIFY DATA Z1 SET COMPLETE
C          (I*4)  IPRTR()   = PARENT INDICES IN DATA SET
C          (I*4)  IGRDR()   = GROUND INDICES IN DATA SET
C          (I*4)  LCK       = MUST BE GREATER THAN 'ITMAXD' & 'IDMAXD'
C                             & 'ITMAX' - ARRAY SIZE FOR SPLINE CALCS.
C          (R*8)  A()       = GENERAL ARRAY
C          (R*8)  DRCOF0(,) = INTERPOLATION OF DRCOFD(,,) W.R.T DTEV()
C          (L*8)  LEXIST    = TRUE --- FILE TO OPEN EXISTS ELSE NOT
C          (I*4)  L1      = PARAMETER = 1
C          (I*4)  IOPT    = DEFINES THE BOUNDARY DERIVATIVES FOR THE
```

```
C                              SPLINE ROUTINE 'XXSPLN', SEE 'XXSPLN'.
C          (L*4)  LSETX   = .TRUE.  => SET UP SPLINE PARAMETERS RELATING
C                                      TO X-AXIS.
C                           .FALSE. => DO NOT SET UP SPLINE PARAMETERS
C                                      RELATING TO X-AXIS.
C                                      (I.E. THEY WERE SET IN A PREVIOUS
C                                            CALL )
C                           (VALUE SET TO .FALSE. BY 'XXSPLN')
C          (R*8)  DY()    = SPLINE INTERPOLATED DERIVATIVES
C
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4UNIT     ADAS      FETCH UNIT NUMBER FOR OUTPUT OF MESSAGES
C          I4FCTN     ADAS      CONVERT STRING TO INTEGER FORM
C
C          (R*8 ADAS FUNCTION - 'R8FUN1' ( X -> X) )
C
C AUTHOR : H. P. SUMMERS, JET
C          K1/1/57
C          JET  EXT. 4941
C
C DATE   : 24/04/94
C
C UPDATE : 21/07/94 - HPS - BYPASS CHECK ON CHARGE STATE COMPLETENESS
C                           FOR XCD AND QCD FILES
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 08-11-95
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST RELEASE
C
C VERSION: 1.2                          DATE: 22-11-95
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - CHANGED TEST FOR ION LIMITS SLIGHTLY FROM
C                 IZ1MIN.GE.IZ1MAX TO IZ1MIN.GT.IZ1MAX TO ALLOW
C                 RUNS FOR HYDROGEN TO PROCEED.
C
C VERSION: 1.3 DATE: 13-10-99
C MODIFIED: Martin O'Mullane
C - PRB definition has been changed and they are now
C                 summed over the parents. This necessitates accessing
C                 the data more like PLT/PRC/PLS than the others.
C               - DSNOLD made same size as DSNINC
C
C VERSION: 1.4 DATE: 02-01-2001
C MODIFIED: Martin O'Mullane
C - Error in logic reading number of metastables in
C                 resolved datasets - change IZ1MAX+IZ1MAX-2.GT.16
C                 to IZ1MAX+IZ1MIN-2.GT.16
C
C VERSION: 1.4 DATE: 24-07-2001
C MODIFIED: Richard Martin
C Added check for whitespace (line 323), so that it can cope
C with SCCS data.
C
C VERSION: 1.5 DATE: 22-12-2008
C MODIFIED: Allan Whiteford
C SAVEd IZ1MAX, IGRDR, IPRTR in addition to the
```

```
C               previously saved variables.
C
C-----------------------------------------------------------------------
      CHARACTER*120       DSNINC
      INTEGER             ICLASS,      IDMAXD,      IFAIL,       IGRDD
      INTEGER             IPRTD,       ISDIMD,      ISMAXD,      ITDIMD
      INTEGER             ITMAX,       ITMAXD,      IZ0,         IZ1
      INTEGER             IZDIMD,      IZMAXD,      NPART(IZDIMD)
      INTEGER             NPARTR(IZDIMD)
      LOGICAL             LPART
      REAL*8              DDENS(ITMAX),             DDENSD(ITDIMD)
      REAL*8              DRCOFD(ISDIMD,ITDIMD,ITDIMD)
      REAL*8              DRCOFI(ITMAX),            DTEV(ITMAX)
      REAL*8              DTEVD(ITDIMD),            ZDATA(ISDIMD)
```

5.88 dxspl1: Subroutine dxspl1 from library adas4xx

```
      SUBROUTINE DXSPL1( ISWIT  , LSWIT  , IZ1   ,
     &                   NDOUT  , NTOUT ,
     &                   NDIN   , NTIN  , NZIN   ,
     &                   IDE    , ITE   , IZE    ,
     &                            TIN   , ZIPT   , EIA    ,
     &                   AIPT   ,
     &                   ZINTRP ,
     &                   ATTY
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: DXSPL1 *********************
C
C  PURPOSE: PERFORMS THE FIRST PART OF A THREE WAY SPLINE ON INPUT DATA.
C           GENERATES A TABLE OF LOG10(SCALED COEF/PWRS) COVERING 'ITE'
C           TEMPERATURES AND 'IDE' DENSITIES FOR THE ELEMENT RECOMBINING
C           ION CHARGE GIVEN BY 'IZ1'.
C
C  CALLING PROGRAM: D1SPLN/D4DATA
C
C  DATA:
C
C           THE  SOURCE  DATA IS CONTAINED AS  MEMBERS  OF  PARTITIONED
C           DATA SETS AS FOLLOWS:
C
C             1. JETUID.ACD<YR>.DATA
C             2. JETUID.SCD<YR>.DATA
C             3. JETUID.CCD<YR>.DATA
C             4. JETUID.PRB<YR>.DATA
C             5. JETUID.PRC<YR>.DATA
C             6. JETUID.PRB<YR>.DATA
C             7. JETUID.PRC<YR>.DATA
C             8. JETUID.PLT<YR>.DATA
C             9. JETUID.PLS<YR>.DATA
C            10. JETUID.MET<YR>.DATA
C
C           WHERE <YR> DENOTES TWO INTEGERS FOR THE YEAR SELECTED.
C
C           THE PARTICULAR TYPE OPENED (1-10) IS SELECTED BY 'ISWIT'
C
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  ISWIT   = DATA TYPE SELECTOR (SEE ABOVE) (1 -> 10)
C  INPUT : (L*4)  LSWIT   = .TRUE.  => IONISATION POTENTIALS PRESENT
C                           .FALSE. => IONS. POTENTIALS NOT PRESENT
C  INPUT : (I*4)  IZ1     = OUTPUT - ELEMENT RECOMBINING ION CHARGE
C
C  INPUT : (I*4)  NDOUT   = OUTPUT - MAXIMUM NUMBER OF DENSITIES
C  INPUT : (I*4)  NTOUT   = OUTPUT - MAXIMUM NUMBER OF TEMPERATURES
C
C  INPUT : (I*4)  NDIN    = INPUT  - MAXIMUM NUMBER OF DENSITIES
C  INPUT : (I*4)  NTIN    = INPUT  - MAXIMUM NUMBER OF TEMPERATURES
C  INPUT : (I*4)  NZIN    = INPUT  - MAXIMUM NUMBER OF CHARGE STATES
C
C  INPUT : (I*4)  ITE     = INPUT  - NUMBER OF REDUCED TEMPERATURES
C  INPUT : (I*4)  IDE     = INPUT  - NUMBER OF REDUCED DENSITIES
C  INPUT : (I*4)  IZE     = INPUT  - NUMBER OF CHARGE STATES/RECOMBINING
C                                    ION CHARGE
```

```
C
C  INPUT : (R*8)  TIN()   = INPUT - SET OF 'ITE' REDUCED ELECTRON TEMPS.
C  INPUT : (R*8)  ZIPT()  = INPUT - SET OF 'IZE' RECOMBINING ION CHARGES
C  INPUT : (R*8)  EIA()   = IONISATION POTENTIALS: ()=ION CHARGE
C                           (UNITS: RYDBERGS)
C
C  INPUT : (R*8)  AIPT(,,)= INPUT - COEFFICIENT/POWER ARRAY.
C                           1ST DIMENSION: REDUCED DENSITY ('DENSR()')
C                           2ND DIMENSION: REDUCED TEMPERATURE ('TR()')
C                           3RD DIMENSION: CHARGE STATE ('ZIPT()')
C
C  OUTPUT: (L*4)  ZINTRP(1)= .TRUE.  => 'ATTY(,)' VALUES INTERPOLATED
C                          = .FALSE. => 'ATTY(,)' VALUES EXTRAPOLATED
C
C  OUTPUT: (R*8)  ATTY(,) = WORKING SPACE FOR 3-WAY SPLINE ITERPOLATION
C                           (STORES VALUES OF 'ANS(1)' )
C                           1ST DIMENSION: TEMPERATURE
C                           2ND DIMENSION: DENSITY
C
C          (I*4)  NZDIM1  = PARAMETER = MUST BE EQUAL TO OR GREATER THAN
C                           THE MAXIMUM NUMBER OF INPUT CHARGE STATES
C          (I*4)  L1      = PARAMETER = 1
C
C          (R*8)  TK2ATE  = PARAMETER = EQUATION CONSTANT = 157890
C
C          (I*4)  I4UNIT  = FUNCTION (SEE ROUTINE SECTION BELOW)
C          (I*4)  IT      = ARRAY SUBSCRIPT USED FOR TEMPERATURE VALUES
C          (I*4)  ID      = ARRAY SUBSCRIPT USED FOR DENSITY VALUES
C          (I*4)  IZ      = ARRAY SUBSCRIPT USED FOR CHARGE STATE VALUES
C          (I*4)  JZ      = RECOMBINING ION CHARGE FOR CHARGE STATE 'IZ'
C          (I*4)  IOPT    = DEFINES THE BOUNDARY DERIVATIVES FOR THE
C                           SPLINE ROUTINE 'XXSPLE', SEE 'XXSPLE'.
C                           (VALID VALUES = 0, 1, 2, 3)
C
C          (L*4)  LSETX   = .TRUE.  => SET UP SPLINE PARAMETERS RELATING
C                                      TO 'XIN' AXIS.
C                           .FALSE. => DO NOT SET UP SPLINE PARAMETERS
C                                      RELATING TO 'XIN' AXIS.
C                                      (I.E. THEY WERE SET IN A PREVIOUS
C                                            CALL )
C                           (VALUE SET TO .FALSE. BY 'XXSPLE')
C
C          (R*8)  ATE     = 'TK2ATE' / INPUT REDUCED TEMPERATURE
C          (R*8)  Z2      = 'JZ' SQUARED
C          (R*8)  Z3      = 'JZ' CUBED
C          (R*8)  ZL      = LOG10 OF CURRENT CHARGE STATE
C          (R*8)  Y       = SCALED 'AIPT(,,)' VALUE
C          (R*8)  Z1(1)   = IZ1
C          (R*8)  ANS(1)  = SPLINE INTERPOLATED LOG10(SCALED 'AIPT(,,)')
C                           VALUE FOR A RECOMBINING ION CHARGE EQUAL TO
C                           'IZ1' AT FIXED TEMPERATURE AND DENSITY.
C          (R*8)  F()     = LOG10 ( 'Y' ) - DIMENSION => CHARGE STATE
C          (R*8)  DF()    = SPLINE INTERPOLATED DERIVATIVES
C
C NOTE:
C                         SPLINE IS CARRIED OUT ON:
C           LOG10(SCALED 'AIPT(,,)' VALUES AT FIXED TEMP AND DENSITY)
C                                 VERSUS
C                           RECOMBINING ION CHARGE
C
C ROUTINES:
```

```
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          XXSPLE     ADAS      SPLINE SUBROUTINE (WITH EXTRAP. INFO)
C          I4UNIT     ADAS      INTEGER*4 FUNCTION    -
C                               FETCH UNIT NUMBER FOR OUTPUT OF MESSAGES
C          R8FUN2     ADAS      REAL*8 FUNCTION: ( X -> 1/(1+X) )
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/37
C          JET EXT. 2520
C
C DATE  :  13/06/91 - PE BRIDEN: ADAS91 VERSION OF 'D4SPL1'
C DATE  :  15/12/92 - PE BRIDEN: IF IZE<=1 THEN ANS(1)=F(1) INSTEAD
C                                OF ANS(1)=F(IZ) (WHICH IN ERROR GIVES
C                                ANS(1)=F(2) IF IZE=1.)
C
C UPDATE:  12/08/93 - HP SUMMERS: INCLUDE ISWIT IN PARAMETER AND ALLOW
C                                 SEPARATE TREATMENT ACCORDING TO ISWIT
C UPDATE:  18/08/93 - HP SUMMERS: ALTER ORDER TO TAKE LOG10 FIRST BEFORE
C                                 ADJUSTING FOR DATA CLASS
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 06-09-95
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST RELEASE
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      INTEGER             IDE,         ISWIT,       ITE,         IZ1
      INTEGER             IZE,         NDIN,        NDOUT,       NTIN
      INTEGER             NTOUT,       NZIN
      LOGICAL             LSWIT,       ZINTRP(1)
      REAL*8              AIPT(NDIN,NTIN,NZIN),     ATTY(NTOUT,NDOUT)
      REAL*8              EIA(50),     TIN(NTIN),   ZIPT(NZIN)
```

5.89 dxspl2: Subroutine dxspl2 from library adas4xx

```
      SUBROUTINE DXSPL2( ISWIT  , LSWIT , IZ1   ,
     &                   NDOUT  , NTOUT ,
     &                   NDIN   ,
     &                   IDE    , ITE   ,
     &                   MAXD   , DIN   , DOUT  ,
     &                   DINTRP ,
     &                   ATTY
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: DXSPL2 *********************
C
C  PURPOSE: PERFORMS THE SECOND PART OF A 3 WAY SPLINE ON INPUT DATA.
C           GENERATES A TABLE OF LOG10( COEFFTS./POWERS )    COVERING
C           'ITE' TEMPERATURES AND 'MAXD' DENSITIES FOR THE ELEMENT
C           RECOMBINING ION CHARGE GIVEN BY 'IZ1'.
C
C  CALLING PROGRAM: D1SPLN/D4DATA
C
C  DATA:
C
C           THE  SOURCE  DATA ORIGINATES AS  MEMBERS  OF  PARTITIONED
C           DATA SETS AS FOLLOWS:
C
C             1. JETUID.ACD<YR>.DATA
C             2. JETUID.SCD<YR>.DATA
C             3. JETUID.CCD<YR>.DATA
C             4. JETUID.PRB<YR>.DATA
C             5. JETUID.PRC<YR>.DATA
C             6. JETUID.PRB<YR>.DATA
C             7. JETUID.PRC<YR>.DATA
C             8. JETUID.PLT<YR>.DATA
C             9. JETUID.PLS<YR>.DATA
C            10. JETUID.MET<YR>.DATA
C
C           WHERE <YR> DENOTES TWO INTEGERS FOR THE YEAR SELECTED.
C
C           THE PARTICULAR TYPE OPENED (1-10) IS SELECTED BY 'ISWIT'
C           IT IS PASSED IN A MODIFIED FORM AFTER PROCESSING BY
C           DXSPL1.
C
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  ISWIT   = DATA TYPE SELECTOR (1 -> 10)
C  INPUT : (L*4)  LSWIT   = .TRUE.  => IONISATION POTENTIALS PRESENT
C                           .FALSE. => IONS. POTENTIALS NOT PRESENT
C  INPUT : (I*4)  IZ1     = OUTPUT - ELEMENT RECOMBINING ION CHARGE
C
C  INPUT : (I*4)  NDOUT   = OUTPUT - MAXIMUM NUMBER OF DENSITIES
C  INPUT : (I*4)  NTOUT   = OUTPUT - MAXIMUM NUMBER OF TEMPERATURES
C
C  INPUT : (I*4)  NDIN    = INPUT  - MAXIMUM NUMBER OF DENSITIES
C
C  INPUT : (I*4)  IDE     = INPUT  - NUMBER OF REDUCED DENSITIES
C  INPUT : (I*4)  ITE     = INPUT  - NUMBER OF REDUCED TEMPERATURES
C
C  INPUT : (I*4)  MAXD    = OUTPUT - NUMBER OF REDUCED DENSITIES
C                                                          ( <= NDOUT )
```

```
C  INPUT : (R*8)  DIN()   = INPUT  - SET OF 'IDE' REDUCED ELECTRON DENS-
C                             ITIES ).
C  INPUT : (R*8)  DOUT()  = OUTPUT - SET OF 'MAXD' ELECTRON  DENSITIES
C                             (UNITS: CM-3).
C
C  OUTPUT: (L*4)  DINTRP() = .TRUE.  => 'ATTY(,)' VALUE FOR DENSITY
C                                       INDEX INTERPOLATED.
C                          = .FALSE. => 'ATTY(,)' VALUE FOR DENSITY
C                                       INDEX EXTRAPOLATED.
C                            1ST DIMENSION: DENSITY INDEX
C
C  IN/OUT: (R*8)  ATTY(,) = WORKING SPACE FOR 3-WAY SPLINE ITERPOLATION
C                           (STORES LOG10 (INTERPOLATED VALUES))
C                           INPUT 'ATTY' VALUES ARE ASSIGNED TO 'YIN' &
C                           THEN 'YOUT' VALUES ARE ASSIGNED TO 'ATTY'.
C                           1ST DIMENSION: TEMPERATURE
C                           2ND DIMENSION: DENSITY
C
C          (I*4)  NDDIM1  = PARAMETER = MUST BE EQUAL TO OR GREATER THAN
C                           THE MAXIMUM NUMBER OF INPUT DENSITIES.
C          (I*4)  NDDIM2  = PARAMETER = MUST BE EQUAL TO OR GREATER THAN
C                           THE MAXIMUM NUMBER OF OUTPUT DENSITIES.
C
C          (I*4)  IT      = ARRAY SUBSCRIPT USED FOR TEMPERATURE VALUES
C          (I*4)  ID      = ARRAY SUBSCRIPT USED FOR DENSITY VALUES
C          (I*4)  IOPT    = DEFINES THE BOUNDARY DERIVATIVES FOR THE
C                           SPLINE ROUTINE 'XXSPLE', SEE 'XXSPLE'.
C                           (VALID VALUES = 0, 1, 2, 3)
C
C          (L*4)  LSETX   = .TRUE.  => SET UP SPLINE PARAMETERS RELATING
C                                      TO 'XIN' AXIS.
C                           .FALSE. => DO NOT SET UP SPLINE PARAMETERS
C                                      RELATING TO 'XIN' AXIS.
C                                      (I.E. THEY WERE SET IN A PREVIOUS
C                                            CALL )
C                           (VALUE SET TO .FALSE. BY 'XXSPLE')
C
C          (R*8)  Z1R7    = 1 / (IZ1**7)
C          (R*8)  YIN()   ='ATTY(,)' AT FIXED TEMPERATURE -
C                                                 DIMENSION => DENSITY
C          (R*8)  DF()    = SPLINE INTERPOLATED DERIVATIVES
C          (R*8)  XOUT()  = 'DOUT()' VALUES CONVERTED TO REDUCED DENSITY
C          (R*8)  YOUT()  = SPLINE INTERPOLATED 'ATTY(,,)' VALUES  FOR
C                           REDUCED DENSITY EQUAL  TO  'XOUT()'  AT  A
C                           FIXED TEMPERATURE.
C
C NOTE:
C                       SPLINE IS CARRIED OUT ON:
C                'ATTY(,,)' VALUES AT FIXED TEMPERATURE
C                                VERSUS
C                         LOG10( REDUCED DENSITY )
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          XXSPLE     ADAS      SPLINE SUBROUTINE (WITH EXTRAP. INFO)
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/37
C          JET EXT. 2520
C
```

```
C DATE  :  13/06/91 - PE BRIDEN: ADAS91 VERSION OF 'D4SPL2'
C
C UPDATE:  07/08/91 - PE BRIDEN: CHANGED THE LINE -
C                                Z1R7 = 1.0 / DBLE(IZ1**7)
C                                TO -
C                                Z1R7 = 1.0 / DBLE(IZ1)**7
C                                TO AVOID INTEGER OVERFLOW WHEN IZ1>21
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 06-09-95
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST RELEASE
C VERSION: 1.2                          DATE: 25-10-97
C MODIFIED: LORNE HORTON (JET)
C               - FORCED ZERO LOW DENSITY DEPENDENCE
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      INTEGER             IDE,         ISWIT,       ITE,         IZ1
      INTEGER             MAXD,        NDIN,        NDOUT,       NTOUT
      LOGICAL             DINTRP(NDOUT),            LSWIT
      REAL*8              ATTY(NTOUT,NDOUT),        DIN(NDIN)
      REAL*8              DOUT(NDOUT)
```

5.90 dxspl3: Subroutine dxspl3 from library adas4xx

```
      SUBROUTINE DXSPL3( ISWIT  , LSWIT , IZ1   ,
     &                   NDOUT  , NTOUT ,
     &                            NTIN  ,
     &                   MAXD   , ITE   ,
     &                   MAXT   , TIN   , TOUT  , EIAVAL ,
     &                   TINTRP ,
     &                   ATTY
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: DXSPL3 *********************
C
C  PURPOSE: PERFORMS THE THIRD PART OF A THREE WAY SPLINE ON INPUT DATA.
C           GENERATES  A  TABLE  OF  LOG10( COEFFTS./POWERS )  COVERING
C           'MAXT' TEMPERATURES AND 'MAXD' DENSITIES FOR THE ELEMENT
C           RECOMBINING ION CHARGE GIVEN BY 'IZ1'.
C
C  CALLING PROGRAM: D1SPLN/D4DATA
C  DATA:
C
C           THE  SOURCE  DATA ORIGINATES AS MEMBERS  OF  PARTITIONED
C           DATA SETS AS FOLLOWS:
C
C             1. JETUID.ACD<YR>.DATA
C             2. JETUID.SCD<YR>.DATA
C             3. JETUID.CCD<YR>.DATA
C             4. JETUID.PRB<YR>.DATA
C             5. JETUID.PRC<YR>.DATA
C             6. JETUID.PRB<YR>.DATA
C             7. JETUID.PRC<YR>.DATA
C             8. JETUID.PLT<YR>.DATA
C             9. JETUID.PLS<YR>.DATA
C            10. JETUID.MET<YR>.DATA
C
C           WHERE <YR> DENOTES TWO INTEGERS FOR THE YEAR SELECTED.
C
C           THE PARTICULAR TYPE OPENED (1-10) IS SELECTED BY 'ISWIT'
C           IT IS PASSED IN A MODIFIED FORM AFTER PROCESSING BY
C           DXSPL1 AND DXSPL2.
C
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  ISWIT   = DATA TYPE SELECTOR (1 -> 8)
C  INPUT : (L*4)  LSWIT   = .TRUE.  => IONISATION POTENTIALS PRESENT
C                           .FALSE. => IONS. POTENTIALS NOT PRESENT
C  INPUT : (I*4)  IZ1     = OUTPUT - ELEMENT RECOMBINING ION CHARGE
C
C  INPUT : (I*4)  NDOUT   = OUTPUT - MAXIMUM NUMBER OF DENSITIES
C  INPUT : (I*4)  NTOUT   = OUTPUT - MAXIMUM NUMBER OF TEMPERATURES
C
C  INPUT : (I*4)  NTIN    = INPUT  - MAXIMUM NUMBER OF TEMPERATURES
C
C  INPUT : (I*4)  MAXD    = INPUT  - NUMBER OF REDUCED DENSITIES
C  INPUT : (I*4)  ITE     = INPUT  - NUMBER OF REDUCED TEMPERATURES
C
C  INPUT : (I*4)  MAXT    = OUTPUT - NUMBER OF REDUCED TEMPERATURES
C                                                        ( <= NTOUT )
C  INPUT : (R*8)  TIN()   = INPUT  - SET OF 'ITE' REDUCED ELECTRON TEMPS
```

```
C  INPUT : (R*8)  TOUT()  = OUTPUT - SET OF 'MAXT' ELECTRON TEMPERATURES
C                                  (UNITS: KELVIN).
C  INPUT : (R*8)  EIAVAL  = IONISATION POTENTIAL (RYDBERGS) FOR
C                           THE ION CHARGE GIVEN BY 'IZ1'.
C
C  OUTPUT: (L*4)  TINTRP() = .TRUE.  => 'ATTY(,)' VALUE FOR TEMPERATURE
C                                       INDEX INTERPOLATED.
C                          = .FALSE. => 'ATTY(,)' VALUE FOR TEMPERATURE
C                                       INDEX EXTRAPOLATED.
C                            1ST DIMENSION: TEMPERATURE INDEX
C
C  IN/OUT: (R*8)  ATTY(,) = WORKING SPACE FOR 3-WAY SPLINE ITERPOLATION
C                           (STORES LOG10 (INTERPOLATED VALUES))
C                           INPUT 'ATTY' VALUES ARE ASSIGNED TO 'YIN' &
C                           THEN 'YOUT' VALUES ARE ASSIGNED TO 'ATTY'.
C                           1ST DIMENSION: TEMPERATURE
C                           2ND DIMENSION: DENSITY
C
C          (I*4)  NTDIM1  = PARAMETER = MUST BE EQUAL TO OR GREATER THAN
C                           THE MAXIMUM NUMBER OF INPUT TEMPERATURES.
C          (I*4)  NTDIM2  = PARAMETER = MUST BE EQUAL TO OR GREATER THAN
C                           THE MAXIMUM NUMBER OF OUTPUT TEMPERATURES.
C
C          (R*8)  EIACON  = PARAMETER = -68570.7
C
C          (I*4)  IT      = ARRAY SUBSCRIPT USED FOR TEMPERATURE VALUES
C          (I*4)  ID      = ARRAY SUBSCRIPT USED FOR DENSITY VALUES
C          (I*4)  IOPT    = DEFINES THE BOUNDARY DERIVATIVES FOR THE
C                           SPLINE ROUTINE 'XXSPLE', SEE 'XXSPLE'.
C                           (VALID VALUES = 0, 1, 2, 3)
C
C          (L*4)  LSETX   = .TRUE.  => SET UP SPLINE PARAMETERS RELATING
C                                      TO 'XIN' AXIS.
C                           .FALSE. => DO NOT SET UP SPLINE PARAMETERS
C                                      RELATING TO 'XIN' AXIS.
C                                      (I.E. THEY WERE SET IN A PREVIOUS
C                                            CALL )
C                           (VALUE SET TO .FALSE. BY 'XXSPLE')
C
C          (R*8)  Z1R2    = 1.0 / (IZ1**2)
C          (R*8)  SCON1   = SCALING CONSTANT
C          (R*8)  SCON2   = SCALING CONSTANT
C          (R*8)  XOUT()  ='TOUT()' CONVERTED TO REDUCED TEMPERATURE
C          (R*8)  YIN()   ='ATTY(,)' AT FIXED DENSITY -
C                                             DIMENSION => TEMPERATURE
C          (R*8)  DF()    = SPLINE INTERPOLATED DERIVATIVES
C          (R*8)  YOUT()  = SPLINE INTERPOLATED 'ATTY(,,)' VALUES  FOR
C                           REDUCED TEMPERATURE  EQUAL TO 'XOUT()' AT A
C                           FIXED DENSITY.
C
C NOTE:
C                        SPLINE IS CARRIED OUT ON:
C                   'ATTY(,,)' VALUES AT FIXED DENSITY
C                                VERSUS
C                         LOG10( REDUCED DENSITY )
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          XXSPLE     ADAS      SPLINE SUBROUTINE (WITH EXTRAP. INFO)
C          I4UNIT     ADAS      UNIT NUMBER FOR WARNING MESSAGES
```

```
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/37
C          JET EXT. 2520
C
C DATE  :  13/06/91 - PE BRIDEN: ADAS91 VERSION OF 'D4SPL3'
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 06-09-95
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST RELEASE
C VERSION: 1.2                          DATE: 25-10-97
C MODIFIED: LORNE HORTON (JET)
C               - ADDED FURTHER CHECKS ON LOW T EXTRAPOLATIONS
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      INTEGER             ISWIT,       ITE,          IZ1,          MAXD
      INTEGER             MAXT,        NDOUT,        NTIN,         NTOUT
      LOGICAL             LSWIT,       TINTRP(NTOUT)
      REAL*8              ATTY(NTOUT,NDOUT),         EIAVAL
      REAL*8              TIN(NTIN),   TOUT(NTOUT)
```

5.91 fmin: Subroutine fmin from library adas4xx

```
c  To get d1mach, mail netlib
c       send d1mach from core
      double precision function fmin(ax,bx,f,tol)
C-----------------------------------------------------------------
c
c      PURPOSE: Find an approximation x to the point where f(x)
c               attains a minimum on the interval (ax,bx).
c
c  input..
c
c  ax    left endpoint of initial interval
c  bx    right endpoint of initial interval
c  f     function subprogram which evaluates  f(x)  for any  x
c        in the interval  (ax,bx)
c  tol   desired length of the interval of uncertainty of the final
c        result (.ge.0.)
c
c  output..
c
c  fmin  abcissa approximating the point where  f  attains a
c        minimum
c
c      the method used is a combination of  golden  section  search  and
c  successive parabolic interpolation.  convergence is never much slower
c  than  that  for  a  fibonacci search.  if  f  has a continuous second
c  derivative which is positive at the minimum (which is not  at  ax  or
c  bx),  then  convergence  is  superlinear, and usually of the order of
c  about  1.324....
c      the function  f  is never evaluated at two points closer together
c  than  eps*abs(fmin)+(tol/3), where eps is  approximately  the  square
c  root  of  the  relative  machine  precision.  if  f  is a unimodal
c  function and the computed values of  f  are  always  unimodal  when
c  separated  by  at least  eps*abs(x)+(tol/3), then  fmin  approximates
c  the abcissa of the global minimum of  f  on the interval  ax,bx  with
c  an error less than  3*eps*abs(fmin)+tol.  if  f  is  not  unimodal,
c  then fmin may approximate a local, but perhaps non-global, minimum to
c  the same accuracy.
c      this function subprogram is a slightly modified  version  of  the
c  algol  60 procedure  localmin  given in richard brent, algorithms for
c  minimization without derivatives, prentice-hall, inc. (1973).
c
C PUT INTO ADAS BY:
C    WILLIAM OSBORN, TESSELLA SUPPORT SERVICES PLC.
C
C DATE:    25TH APRIL 1996
C
C VERSION: 1.1 DATE: 25-04-96
C MODIFIED: WILLIAM OSBORN
C     - FOUND AT WWW.NETLIB.ORG
C
C VERSION: 1.2 DATE: 20-09-99
C MODIFIED: RICHARD MARTIN
C CHANGED FROM fmin.f TO fmin.for
C
C VERSION: 1.3 DATE: 16-05-07
C MODIFIED: Allan Whiteford
C     - Updated comments as part of subroutine documentation
C             procedure.
```

```
C
C------------------------------------------------------------------
      double precision ax,bx,f,tol
      double precision  a,b,c,d,e,eps,xm,p,q,r,tol1,t2,u,v,w,fu,fv,fw,
     2    fx,x,tol3
c      double precision  dabs,dsqrt,d1mach
      double precision  dabs,dsqrt,dpmpar
c
c  c is the squared inverse of the golden ratio
      c=0.5d0*(3.0d0-dsqrt(5.0d0))
c
c  eps is approximately the square root of the relative machine
c  precision.
c
C W.R.O. CHANGED FOLLOWING LINE FROM 10 eps=d1mach(4)

   10 eps=dpmpar(1)
      tol1=eps+1.0d0
      eps=dsqrt(eps)
c
      a=ax
      b=bx
      v=a+c*(b-a)
      w=v
      x=v
      e=0.0d0
      fx=f(x)
      fv=fx
      fw=fx
      tol3=tol/3.0d0
c
c  main loop starts here
c
   20 xm=0.5d0*(a+b)
      tol1=eps*dabs(x)+tol3
      t2=2.0d0*tol1
c
c  check stopping criterion
c
      if (dabs(x-xm).le.(t2-0.5d0*(b-a))) go to 190
      p=0.0d0
      q=0.0d0
      r=0.0d0
      if (dabs(e).le.tol1) go to 50
c
c  fit parabola
c
      r=(x-w)*(fx-fv)
      q=(x-v)*(fx-fw)
      p=(x-v)*q-(x-w)*r
      q=2.0d0*(q-r)
      if (q.le.0.0d0) go to 30
      p=-p
      go to 40
   30 q=-q
   40 r=e
      e=d
   50 if ((dabs(p).ge.dabs(0.5d0*q*r)).or.(p.le.q*(a-x))
     2          .or.(p.ge.q*(b-x))) go to 60
c
c  a parabolic-interpolation step
```

```
c
      d=p/q
      u=x+d
c
c  f must not be evaluated too close to ax or bx
c
      if (((u-a).ge.t2).and.((b-u).ge.t2)) go to 90
      d=tol1
      if (x.ge.xm) d=-d
      go to 90
c
c  a golden-section step
c
   60 if (x.ge.xm) go to 70
      e=b-x
      go to 80
   70 e=a-x
   80 d=c*e
c
c  f must not be evaluated too close to x
c
   90 if (dabs(d).lt.tol1) go to 100
      u=x+d
      go to 120
  100 if (d.le.0.0d0) go to 110
      u=x+tol1
      go to 120
  110 u=x-tol1
  120 fu=f(u)
c
c  update  a, b, v, w, and x
c
      if (fx.gt.fu) go to 140
      if (u.ge.x) go to 130
      a=u
      go to 140
  130 b=u
  140 if (fu.gt.fx) go to 170
      if (u.ge.x) go to 150
      b=x
      go to 160
  150 a=x
  160 v=w
      fv=fw
      w=x
      fw=fx
      x=u
      fx=fu
      go to 20
  170 if ((fu.gt.fw).and.(w.ne.x)) go to 180
      v=w
      fv=fw
      w=u
      fw=fu
      go to 20
  180 if ((fu.gt.fv).and.(v.ne.x).and.(v.ne.w)) go to 20
      v=u
      fv=fu
      go to 20
c
c  end of main loop
```

```
c
  190 fmin=x
      return
      end
      DOUBLE PRECISION    AX,          BX,          TOL
```

5.92 gpcalc: Subroutine gpcalc from library adas4xx

```
       REAL*8 FUNCTION GPCALC(CFAC1)
       IMPLICIT REAL*8(A-H,O-Z)
C-----------------------------------------------------------------------
C  PURPOSE: ROUTINE TO PROVIDE BURGESS GENERAL PROGRAM RESULTS AT
C  A SERIES OF TEMPERATURES AND AT ZERO DENSITY.
C
C  EQUAL THE GENERAL FORMULA RESULTS AS FAR AS POSSIBLE BY MODIFICATION
C  OF BETHE CORRECTIONS VIA A SINGLE SCALING PARAMETER CORFAC.
C   THE CORRECTION FACTORS USED IN THE GENERAL PROGRAM
C  ARE OBTAINED BY ADJUSTMENT OF STANDARD SETS FOR SPECIFIC TYPES OF
C  TRANSITION.  THE ADJUSTMENT IS
C    (NEW COR(J))=EXP(-CORFAC/(L**DF+0.5))*(STANDARD COR(J)
C  THE STANDARD COR'S ARE AS FOLLOWS:
C   TYPE    TRANSITION                    COR'S                      DF
C     1   NI=1,NJ>=2,LJ=LI+1:       0.05,0.30,0.50,0.90             2.0
C     2   NI=2,NJ=3,LJ=LI+1:        0.01,0.02,0.20,0.40,0.70,0.90   1.0
C     3   NI=2,NJ=3,LJ=LI-1:        0.01,0.01,0.01,0.08,0.30,0.70   1.0
C     4   NJ-NI=0, LJ=LI+1 :        0.30,0.35,0.40,0.45,0.70,0.90   0.5
C     5   NJ-NI=0, LJ=LI-1 :        0.30,0.35,0.40,0.45,0.70,0.90   0.5
C     6   NJ-NI>0, LJ=LI+1 :        0.01,0.02,0.20,0.40,0.70,0.90   1.0
C     7   NJ-NI>0, LJ=LI-1 :        0.01,0.01,0.01,0.08,0.30,0.70   1.0
C
C     (1)  INCLUDE NCUT AND EXTEND ARRAY SIZES
C     (2)  IMPLIMENTATION OF NCUT,LOW TEMPERATURE CHECK, CORRECTION
C          INVOLVING V1
C
C  ********** H.P. SUMMERS, JET             11 JUNE 1987  *************
C  ********** H.P. SUMMERS, JET     MOD.(1) 24 AUG  1989  *************
C  ********** W.J. DICKSON, JET     MOD.(2) 14 DEC  1989  *************
C  INPUT
C      MAXT=NUMBER OF TEMPERATURES
C      TEA(I)=ELECTRON TEMPERATURES (K)
C      Z1=RECOMBINING ION CHARGE
C      N0=LOWEST ACCESSIBLE N-SHELL BY RECOMBINATION
C      V0=EFFECTIVE PRINCIPAL QUANTUM NUMBER OF LOWEST ACCESSIBLE SHELL
C      NI=LOWER PRINCIPAL QUANTUM NUMBER OF PARENT TRANSITION
C      LI=LOWER ANGULAR QUANTUM NUMBER OF PARENT TRNASITION.
C      WI=LOWER PARENT STATE STATISTICAL WEIGHT.
C      NJ=UPPER PRINCIPAL QUANTUM NUMBER OF PARENT TRANSITION
C      LJ=UPPER ANGULAR QUANTUM NUMBER OF PARENT TRNASITION.
C      WJ=UPPER PARENT STATE STATISTICAL WEIGHT.
C      EIJ=PARENT TRANSITION ENERGY (RYD)
C      FIJ=ABSORPTION OSCILLATOR STRENGTH OF PARENT TRANSITION
C      EDISPG=UNIFORM ENERGY DISPLACEMENT FOR GENERAL FORMULA
C      SCALEG=UNIFORM SCALING OF GENERAL FORMULA
C      PHFRAC=INITIAL ESTIMATE OF PHASE SPACE FACTOR
C      CORFAC=INITIAL ESTIMATE OF BETHE CORRECTION SCALER
C      NCUT  =HIGH N CUT-OFF (APPLICABLE TO METASTABLE INITIAL STATES)
C  OUTPUT
C      ALFO(I)=GENERAL PROGRAM DIELECTRONIC COEFFICIENTS (CM+3 SEC-1)
C      PHFRAC=REVISED PHASE SPACE FACTOR
C      CORFAC=REVISED BETHE CORRECTION SCALER
C
C  UPDATE:  07/03/96  HP SUMMERS - INCREASED NRAT FROM 15 TO 100
C
C UNIX-IDL PORT:
C    WILLIAM OSBORN, TESSELLA SUPPORT SERVICES PLC.
C
```

```
C DATE:    19TH APRIL 1996
C
C VERSION: 1.1 DATE: 19-04-96
C MODIFIED: WILLIAM OSBORN
C     - NRAT WAS 200 FOR SOME REASON : REDUCED TO 100
C
C VERSION: 1.2 DATE: 25-04-96
C MODIFIED: WILLIAM OSBORN
C     - CHANGED GPCALC TO A FUNCTION FOR USE BY FMIN
C
C VERSION : 1.3
C DATE    : 06-01-2004
C MODIFIED: Martin O'Mullane
C            - Old IBM statement labels in columns 73-80 removed.
C
C VERSION : 1.4
C DATE    : 16-05-2007
C MODIFIED: Allan Whiteford
C            - Updated comments as part of subroutine documentation
C              procedure.
C
C-----------------------------------------------------------------------
      REAL*8              CFAC1
```

5.93 gpcalcx: Subroutine gpcalcx from library adas4xx

```
       REAL*8 FUNCTION GPCALCX(XV0)
       IMPLICIT REAL*8(A-H,O-Z)
C-----------------------------------------------------------------------
C  PURPOSE: ROUTINE TO PROVIDE BURGESS GENERAL PROGRAM RESULTS AT A
C  SERIES OF TEMPERATURES AND AT ZERO DENSITY.
C
C  EQUAL THE GENERAL FORMULA RESULTS AS FAR AS POSSIBLE BY MODIFICATION
C  OF BETHE CORRECTIONS VIA A SINGLE SCALING PARAMETER CORFAC.
C   THE CORRECTION FACTORS USED IN THE GENERAL PROGRAM
C  ARE OBTAINED BY ADJUSTMENT OF STANDARD SETS FOR SPECIFIC TYPES OF
C  TRANSITION.  THE ADJUSTMENT IS
C    (NEW COR(J))=EXP(-CORFAC/(L**DF+0.5))*(STANDARD COR(J)
C  THE STANDARD COR'S ARE AS FOLLOWS:
C   TYPE    TRANSITION                    COR'S                      DF
C     1   NI=1,NJ>=2,LJ=LI+1:       0.05,0.30,0.50,0.90             2.0
C     2   NI=2,NJ=3,LJ=LI+1:        0.01,0.02,0.20,0.40,0.70,0.90   1.0
C     3   NI=2,NJ=3,LJ=LI-1:        0.01,0.01,0.01,0.08,0.30,0.70   1.0
C     4   NJ-NI=0, LJ=LI+1 :        0.30,0.35,0.40,0.45,0.70,0.90   0.5
C     5   NJ-NI=0, LJ=LI-1 :        0.30,0.35,0.40,0.45,0.70,0.90   0.5
C     6   NJ-NI>0, LJ=LI+1 :        0.01,0.02,0.20,0.40,0.70,0.90   1.0
C     7   NJ-NI>0, LJ=LI-1 :        0.01,0.01,0.01,0.08,0.30,0.70   1.0
C
C     (1)  INCLUDE NCUT AND EXTEND ARRAY SIZES
C     (2)  IMPLIMENTATION OF NCUT,LOW TEMPERATURE CHECK, CORRECTION
C          INVOLVING V1
C     (3)  NRAT INCREASED FROM 10 TO 15
C
C  ********** H.P. SUMMERS, JET             11 JUNE 1987  *************
C  ********** H.P. SUMMERS, JET     MOD.(1) 24 AUG  1989  *************
C  ********** W.J. DICKSON, JET     MOD.(2) 14 DEC  1989  *************
C  ********** P.E. BRIDEN ,TESSELLA MOD.(3) 23 AUG  1994  *************
C  INPUT
C      MAXT=NUMBER OF TEMPERATURES
C      TEA(I)=ELECTRON TEMPERATURES (K)
C      Z1=RECOMBINING ION CHARGE
C      N0=LOWEST ACCESSIBLE N-SHELL BY RECOMBINATION
C      V0=EFFECTIVE PRINCIPAL QUANTUM NUMBER OF LOWEST ACCESSIBLE SHELL
C      NI=LOWER PRINCIPAL QUANTUM NUMBER OF PARENT TRANSITION
C      LI=LOWER ANGULAR QUANTUM NUMBER OF PARENT TRNASITION.
C      WI=LOWER PARENT STATE STATISTICAL WEIGHT.
C      NJ=UPPER PRINCIPAL QUANTUM NUMBER OF PARENT TRANSITION
C      LJ=UPPER ANGULAR QUANTUM NUMBER OF PARENT TRNASITION.
C      WJ=UPPER PARENT STATE STATISTICAL WEIGHT.
C      EIJ=PARENT TRANSITION ENERGY (RYD)
C      FIJ=ABSORPTION OSCILLATOR STRENGTH OF PARENT TRANSITION
C      EDISPG=UNIFORM ENERGY DISPLACEMENT FOR GENERAL FORMULA
C      SCALEG=UNIFORM SCALING OF GENERAL FORMULA
C      PHFRAC=INITIAL ESTIMATE OF PHASE SPACE FACTOR
C      CORFAC=INITIAL ESTIMATE OF BETHE CORRECTION SCALER
C      NCUT  =HIGH N CUT-OFF (APPLICABLE TO METASTABLE INITIAL STATES)
C  OUTPUT
C      ALFO(I)=GENERAL PROGRAM DIELECTRONIC COEFFICIENTS (CM+3 SEC-1)
C      PHFRAC=REVISED PHASE SPACE FACTOR
C      CORFAC=REVISED BETHE CORRECTION SCALER
C UNIX-IDL PORT:
C    WILLIAM OSBORN, TESSELLA SUPPORT SERVICES PLC.
C
C DATE:    19TH APRIL 1996
```

```
C
C VERSION: 1.1 DATE: 19-04-96
C MODIFIED: WILLIAM OSBORN
C     - NRAT INCREASED FROM 15 TO 100 IN LINE WITH GPCALL
C
C VERSION: 1.2 DATE: 25-04-96
C MODIFIED: WILLIAM OSBORN
C     - CONVERTED TO A FUNCTION FOR FMIN TO USE
C
C VERSION: 1.3 DATE: 16-05-07
C MODIFIED: Allan Whiteford
C     - Old IBM statement labels in columns 73-80 removed.
C     - Updated comments as part of subroutine documentation
C             procedure.
C
C-----------------------------------------------------------------------
      REAL*8              XV0
```

5.94 gpcall: Subroutine gpcall from library adas4xx

```
       SUBROUTINE GPCALL(MAXT,TEA,ALFDAT,ALFO,ALFGF,Z1,N0,V0,NI,LI,WI,
     &NJ,LJ,WJ,EIJ,F,EDISPG,SCALEG,PHFRAC,CORFAC,NCUT,ITYPE,IOPT,
     &IFSEL,IZ0)
       IMPLICIT REAL*8(A-H,O-Z)
C-----------------------------------------------------------------------
C  PURPOSE: PROVIDE BURGESS GENERAL FORMULA RESULTS AT A SERIES OF
C  TEMPERATURES, AND ALSO TO PRODUCE BURGESS GENERAL PROGRAM RESULTS
C  AT  ZERO DENSITY AT THE SAME TEMPERATURES.
C
C  THE LATTER ARE ADJUSTED TO EQUAL INPUT DATA AS FAR AS POSSIBLE BY
C  MODIFICATION OF BETHE CORRECTIONS VIA A SINGLE SCALING PARAMETER
C  CORFAC. THE ROUTINE FMIN   IS USED TO OPTIMISE CORFAC.
C
C  ********** H.P. SUMMERS, JET             11 JUNE 1987  *************
C  ********** H.P. SUMMERS, JET     MOD.(1) 24 AUG  1989  *************
C  ********** W.J. DICKSON, JET     MOD.(2) 14 DEC  1989  *************
C  ********** W.J. DICKSON, JET     MOD.(3)  7 AUG  1990  *************
C  INPUT
C      MAXT=NUMBER OF TEMPERATURES
C      TEA(I)=ELECTRON TEMPERATURES (K)
C      Z1=RECOMBINING ION CHARGE
C      N0=LOWEST ACCESSIBLE N-SHELL BY RECOMBINATION
C      V0=EFFECTIVE PRINCIPAL QUANTUM NUMBER OF LOWEST ACCESSIBLE SHELL
C      NI=LOWER PRINCIPAL QUANTUM NUMBER OF PARENT TRANSITION
C      LI=LOWER ANGULAR QUANTUM NUMBER OF PARENT TRNASITION.
C      WI=LOWER PARENT STATE STATISTICAL WEIGHT.
C      NJ=UPPER PRINCIPAL QUANTUM NUMBER OF PARENT TRANSITION
C      LJ=UPPER ANGULAR QUANTUM NUMBER OF PARENT TRNASITION.
C      WJ=UPPER PARENT STATE STATISTICAL WEIGHT.
C      EIJ=PARENT TRANSITION ENERGY (RYD)
C      FIJ=ABSORPTION OSCILLATOR STRENGTH OF PARENT TRANSITION
C      EDISPG=UNIFORM ENERGY DISPLACEMENT FOR GENERAL FORMULA
C      SCALEG=UNIFORM SCALING OF GENERAL FORMULA
C      PHFRAC=INITIAL ESTIMATE OF PHASE SPACE FACTOR
C      CORFAC=INITIAL ESTIMATE OF BETHE CORRECTION SCALER
C      NCUT  =HIGH N CUT-OFF (APPLICABLE TO METASTABLE INITIAL STATES)
C  OUTPUT
C      ALFO(I)=GENERAL PROGRAM DIELECTRONIC COEFFICIENTS (CM+3 SEC-1)
C      ALFGF(I)=GENERAL FORMULA DIELECTRONIC COEFFICIENTS
C      PHFRAC=REVISED PHASE SPACE FACTOR
C      CORFAC=REVISED BETHE CORRECTION SCALER
C
C  UPDATE:  07/03/96  HP SUMMERS - INCREASED NRAT FROM 15 TO 100

C UNIX-IDL PORT:
C    WILLIAM OSBORN, TESSELLA SUPPORT SERVICES PLC.
C
C DATE:    19TH APRIL 1996
C
C VERSION: 1.1 DATE: 19-04-96
C MODIFIED: WILLIAM OSBORN
C     -NO CHANGES
C
C VERSION: 1.2 DATE: 25-04-96
C MODIFIED: WILLIAM OSBORN
C     -REPLACED CALLS TO E04ABF WITH FUNCTION FMIN FROM WWW.NETLIB
C
C VERSION: 1.3 DATE: 09-08-96
```

```
C MODIFIED: WILLIAM OSBORN
C     -REMOVED COMMA FROM WRITE STATEMENT (WRITING OF XV0)
C
C VERSION: 1.4 DATE: 16-05-07
C MODIFIED: Allan Whiteford
C     - Old IBM statement labels in columns 73-80 removed.
C     - Updated comments as part of subroutine documentation
C             procedure.
C
C-----------------------------------------------------------------------
      INTEGER             IFSEL,       IOPT,        ITYPE,       IZ0
      INTEGER             LI,          LJ,          MAXT,        N0
      INTEGER             NCUT,        NI,          NJ
      REAL*8              ALFDAT(NRAT),             ALFGF(NRAT)
      REAL*8              ALFO(NRAT),  CORFAC,      EDISPG,      EIJ
      REAL*8              F,           PHFRAC,      SCALEG
      REAL*8              TEA(NRAT),   V0,          WI,          WJ
      REAL*8              Z1
```

5.95 init: Subroutine init from library adas4xx

```
      SUBROUTINE INIT(  ELEM, NION, NIONS, IZ,
     &                  SQI  ,SEQ   ,IZ0, SPEC)
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: DCISPF *********************
C
C  PURPOSE: DETRMINATION OF CHARGE OF THE SELECTED ION AND OBTAIN
C SPECTROSCOPIC NOTATION.
C
C  CALLING PROGRAM: ADAS412
C
C  INPUT:
C
C (C*2) ELEM : SELECTED ELEMENT.
C
C (I*4) NIONS : NUMBER OF SELECTED IONS.
C
C     (I*4) IZ(NION): CHARGES OF THE SELECTED IONS.
C
C  OUTPUT:
C
C     (C*2) SQI(NION): ISO-ELECTRONIC SEQUENCE - EG LI.
C
C     (C*7) SEQ(NION): ISO-ELECTRONIC SEQUENCE - EG Li-like.
C
C (I*4) IZ0:        NUCLEAR CHARGE.
C
C     (C*11) SPEC(NIONS):   ION IN SPECTROSCOPIC NOTATION.
C
C AUTHOR: RICHARD MARTIN, UNIVERSITY OF STRATHCLYDE, OCTOBER 1997
C
C VERSION: 1.1 DATE: 27-10-97
C VERSION: 1.2 DATE: 01-12-97
C MODIFIED: RICHARD MARTIN
C - RE-ORDERED VARIABLE DECLARATIONS.
C
C-----------------------------------------------------------------------
C
      CHARACTER*2         ELEM
      CHARACTER*7         SEQ(NION)
      CHARACTER*11        SPEC(NION)
      CHARACTER*2         SQI(NION)
      INTEGER             IZ(NION),    IZ0,         NION,        NIONS
```

5.96 ioadas1: Subroutine ioadas1 from library adas4xx

```
       SUBROUTINE IOADAS1( iunt   , DATE    ,
     &                     NTDIM  , NNEDIM  , NZDIM  , DATA  ,
     &                     ITMAX  , IDMAX   , IZMAX  ,
     &                     DENSE  , TEMPE   ,
     &                     IZ0    , NAME    , METHOD ,
     &                     CSTRG1 , CSTRG2  , LINFO  ,
     &                     user   , type    , dsn03  , dsn35 , lfilter
     &                   )


C-----------------------------------------------------------------------
C
C PURPOSE : TO WRITE ADAS DATA WHICH IS DEPENDENT ON DENSITY
C           AND HAS NO SEPARATE 'NEUTRAL' STAGE.
C
C           THE DATA IS IN THE FORM :-
C                        DATA(IT,ID,IZ)
C           WHERE,
C                  IT     :  TEMPERATURE INDEX ( 1 - ITMAX ) NTDIM
C                  ID     :  DENSITY     INDEX ( 1 - IDMAX ) NNEDIM
C                  IZ     :  STAGE       INDEX ( 1 - ITMAX ) NZDIM
C
C           WITH ELECTRON TEMPERATUES ---- TEMPE(1 - ITMAX)
C                ELECTRON DENSITIES   ---- DENSE(1 - IDMAX)
C
C           (I*4)  iunt    =  STREAM NUMBER (PREVIOUSLY ALLOCATED)
C
C
C           INPUT
C           ~~~~~
C           (R*4)  DATA   :  PROFILE ARRAY (SEE ABOVE)
C           (I*4)  ITMAX  :  NUMBER OF TEMPERATURE INDICIES
C           (I*4)  IDMAX  :  NUMBER OF DENSITY     INDICIES
C           (I*4)  IZMAX  :  NUMBER OF STAGE       INDICIES
C           (R*4)  DENSE  :  ELECTRON DENSITIES
C           (R*4)  TEMPE  :  ELECTRON TEMPERATURES
C           (I*4)  IZ0    :  NUCLEAR CHARGE OF SEPCIES
C           (C*13) NAME   :  NAME OF ELEMENT
C           (C*24) METHOD :  METHOD USED IN THE CALCULATIONS
C           (C*10) CSTRG1 :  FIRST PARENT/GROUND INFORMATION STRING
C           (C*10) CSTRG2 :  SECOND PARENT/GROUND INFORMATION STRING
C           (C*8)  LINFO  :  WAVELENGTH IDENTIFIER FOR SPECIFIC LINE
C
C
C BASED ON IOADAS1 BY
C          JAMES SPENCE
C          JET/TESSELLA SUPPORT SERVICES PLC  23/3/90
C
C MODIFIED TO WRITE ONLY
C          M. O'MULLANE    11/8/92
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 15-04-96
C MODIFIED: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST CONVERTED, CALL TO XXDATE REMOVED
C
C VERSION: 1.2 DATE: 21-07-2003
C MODIFIED: Martin O'Mullane
```

```
C            - Add a comment section at the end.
C            - Remove redundant variables.
C
C VERSION: 1.3 DATE: 17-05-2007
C MODIFIED: Allan Whiteford
C            - Updated comments as part of subroutine documentation
C              procedure.
C
C-----------------------------------------------------------------------
      CHARACTER*10        CSTRG1,      CSTRG2
      CHARACTER*8         DATE
      CHARACTER*80        DSN03,       DSN35
      CHARACTER*8         LINFO(NZDIM)
      CHARACTER*24        METHOD
      CHARACTER*13        NAME
      CHARACTER*40        TYPE
      CHARACTER*30        USER
      INTEGER             IDMAX,       ITMAX,       IUNT,        IZ0
      INTEGER             IZMAX,       NNEDIM,      NTDIM,       NZDIM
      LOGICAL             LFILTER
      REAL*8              DATA(NTDIM,NNEDIM,NZDIM), DENSE(NNEDIM)
      REAL*8              TEMPE(NTDIM)
```

5.97 ioadas2: Subroutine ioadas2 from library adas4xx

```
       SUBROUTINE IOADAS2( iunt   , DATE    ,
     &                     NTDIM  , NNEDIM  , NZDIM  ,
     &                     DATA0  , DATA    ,
     &                     ITMAX  , IDMAX   , IZMAX  ,
     &                     DENSE  , TEMPE   ,
     &                     IZ0    , NAME    , METHOD ,
     &                     CSTRG1 , CSTRG2  , LINFO  ,
     &                     user   , type    , dsn03  , dsn35 , lfilter
     &                   )

C-----------------------------------------------------------------------
C
C PURPOSE : TO WRITE ADAS DATA WHICH IS DEPENDENT ON DENSITY
C           AND HAS SEPARATE 'NEUTRAL' STAGE.
C
C           THE DATA IS IN THE FORM :-
C                       DATA(IT,ID,IZ)
C                       DATA0(IT,ID)
C           WHERE,
C                  IT     :  TEMPERATURE INDEX ( 1 - ITMAX )
C                  ID     :  DENSITY     INDEX ( 1 - IDMAX )
C                  IZ     :  STAGE       INDEX ( 1 - ITMAX )
C
C           WITH ELECTRON TEMPERATUES ---- TEMPE(1 - ITMAX)
C                ELECTRON DENSITIES   ---- DENSE(1 - IDMAX)
C
C           (I*4)  iunt    =  STREAM NUMBER (PREVIOUSLY ALLOCATED)
C
C
C           INPUT
C           ~~~~~
C           (R*4)  DATA   :  PROFILE ARRAY (SEE ABOVE)
C           (R*4)  DATA0  :  PROFILE ARRAY (SEE ABOVE)
C           (I*4)  ITMAX  :  NUMBER OF TEMPERATURE INDICIES
C           (I*4)  IDMAX  :  NUMBER OF DENSITY     INDICIES
C           (I*4)  IZMAX  :  NUMBER OF STAGE       INDICIES
C           (R*4)  DENSE  :  ELECTRON DENSITIES
C           (R*4)  TEMPE  :  ELECTRON TEMPERATURES
C           (I*4)  IZ0    :  NUCLEAR CHARGE OF SEPCIES
C           (C*13) NAME   :  NAME OF ELEMENT
C           (C*25) METHOD :  METHOD USED IN THE CALCULATIONS
C           (C*10) CSTRG1 :  FIRST PARENT/GROUND INFORMATION STRING
C           (C*10) CSTRG2 :  SECOND PARENT/GROUND INFORMATION STRING
C           (C*8)  LINFO  :  WAVELENGTH IDENTIFIER FOR SPECIFIC LINE
C
C
C BASED ON IOADAS2 BY
C          JAMES SPENCE
C          JET/TESSELLA SUPPORT SERVICES PLC  23/3/90
C
C MODIFIED TO WRITE ONLY
C          M. O'MULLANE    11/8/92
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 15-04-96
C MODIFIED: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST CONVERTED
C
```

```
C
C VERSION: 1.2 DATE: 21-07-2003
C MODIFIED: Martin O'Mullane
C            - Add a comment section at the end which requires user, date
C              and adf03/adf35 filenames.
C            - Remove redundant variables.
C
C VERSION: 1.3 DATE: 17-05-2007
C MODIFIED: Allan Whiteford
C           - Updated comments as part of subroutine documentation
C             procedure.
C
C-----------------------------------------------------------------------
      CHARACTER*10        CSTRG1,      CSTRG2
      CHARACTER*8         DATE
      CHARACTER*80        DSN03,       DSN35
      CHARACTER*8         LINFO(NZDIM)
      CHARACTER*24        METHOD
      CHARACTER*13        NAME
      CHARACTER*40        TYPE
      CHARACTER*30        USER
      INTEGER             IDMAX,       ITMAX,       IUNT,        IZ0
      INTEGER             IZMAX,       NNEDIM,      NTDIM,       NZDIM
      LOGICAL             LFILTER
      REAL*8              DATA(NTDIM,NNEDIM,NZDIM), DATA0(NTDIM,NNEDIM)
      REAL*8              DENSE(NNEDIM),            TEMPE(NTDIM)
```

5.98 ionbal: Subroutine ionbal from library adas4xx

```
       SUBROUTINE IONBAL( YEAR  , YEARDF, IFAIL ,
     &                    IZ0   , ITMAX ,
     &                    DTEV  , DDENS , DDENSH,
     &                    FABUND
     &                  )
C-----------------------------------------------------------------------
C  ****************** FORTRAN77 SUBROUTINE: IONBAL *********************
C
C VERSION : 1.1
C
C CALLING PROGRAM: ADAS412
C
C PURPOSE : TO EVALUATE EQUILIBRIUM IONIS. BALANCE IN A PLASMA OF
C           FIXED ELECTRON TEMPERATURE, ELECTRON DENSITY AND NEUTRAL
C           HYDROGEN DENSITY.
C
C NOTE    : ATOMIC RATE COEFFICIENT DATA ARE EXTRACTED FROM THE
C           MASTER ELEMENT FILES USING THE SUBROUTINE 'DHDATA'.
C
C INPUT   : (C*2)  YEAR       = YEAR OF DATA
C           (C*2)  YEARDF     = DEFAULT YEAR OF DATA IF REQUESTED YEAR
C                               DOES NOT EXIST.
C           (I*4)  IZ0        = NUCLEAR CHARGE
C           (I*4)  ITMAX      = NUMBER OF DTEV(),DDENS(),DDENSH() SETS
C           (R*8)  DTEV()     = DLOG10(ELECTRON TEMPERATURES (EV))
C           (R*8)  DDENS()    = DLOG10(ELECTRON DENSITIES (CM-3))
C           (R*8)  DDENSH()   = DLOG10(NEUTRAL. H  DENSITIES (CM-3))
C
C OUTPUT  : (I*4)  IFAIL      = 0     IF ROUTINE SUCCESSFUL
C                               1     IF ROUTINE UNSUCCESSFUL
C           (R*8)  FABUND()   = FRACTIONAL ABUNDANCES FOR DTEV() ETC.
C
C PROGRAM : (C*80) FINFO      = INFORMATION STRING
C           (C*2)  YEARIN     = ACTIVE YEAR OF DATA WHEN CYCLING
C                               (RESET IF NECESSARY FROM YEAR TO YEARDF)
C           (I*4)  NTDIM      = MAXIMUM NUMBER OF TEMP,DENS PAIRS
C           (I*4)  NTDIMD     = MAXIMUM NUMBER OF DATA TEMP & DENS
C           (I*4)  NZDIM      = MAXIMUM NUMBER OF IONISATION STAGES
C           (I*4)  ICLASA()   = CLASSES OF DATA TO BE EXTRACTED
C           (I*4)  NCLASS     = NUMBER OF DATA CLASSES TO BE EXTRACTED
C           (I*4)  ICLASS     = INDEX OF PARTICULAR CLASS
C           (I*4)  IZZ        = ION CHARGE
C           (I*4)  IZ1        = ION CHARGE+1
C           (I*4)  IEVCUT     = ENERGY CUTOFF (EV)
C           (I*4)  ITMAXD     = NUMBER OF DATA DTEVD()
C           (I*4)  IDMAXD     = NUMBER OF DATA DDENS()
C           (I*4)  IZMAXD     = NUMBER OF DATA ZDATA()
C           (I*4)  IT         = INDEX USED WITH TEMPS
C           (I*4)  ICL        = INDEX USED WITH DATA CLASSES
C           (I*4)  N          = NO. OF IONIS. STAGES INCL. BARE NUCLEUS
C                               (EQUALS IZ0+1)
C           (I*4)  IZM        = ION CHARGE -1
C           (R*8)  DENS()     = ELECTRON DENSITIES (CM-3)
C           (R*8)  DENSH()    = NEUTRAL H DENSITIES (CM-3)
C           (R*8)  DTEVD()    = DLOG10(DATA ELECTRON TEMPS (EV))
C           (R*8)  DDENSD()   = DLOG10(DATA ELECTRON DENSITIES (CM-3))
C           (R*8)  ZDATA()    = Z1 CHARGES IN DATA SET
C           (R*8)  DRCOFD(,,) = DLOG10(DATA RATE COEFFICIENTS (CM3/S))
C           (R*8)  DRCOFI()   = INTERPOLATION OF DRCOFD(,,) FOR
```

```
C                                DTEV() & DDENS()
C           (R*8)  ACDA(,)     = INTERPOLATED RECOM. COEFFT (CM3/S)
C           (R*8)  SCDA(,)     = INTERPOLATED IONIS. COEFFT (CM3/S)
C           (R*8)  CCDA(,)     = INTERPOLATED CXR COEFFT. (CM3/S)
C           (R*8)  POPF()      = STAGE FRACTIONAL ABUNDANCES
C           (R*8)  EV          = TEMPERATURE (K) EUIVALENT TO 1 EV
C           (R*8)  RH          = RATIO (H DENS)/(ELEC. DENS)
C           (R*8)  U           = TEMPORARY PARAMETER
C           (R*8)  SUM         = TEMPORARY PARAMETER
C
C ROUTINES:
C           ROUTINE    SOURCE    BRIEF DESCRIPTION
C           ------------------------------------------------------------
C           DHDATA     ADAS    EXTRACT 'SANC0' DATA FROM MASTER FILES
C
C
C AUTHOR  : HP SUMMERS
C           K1/1/57
C           JET EXT. 4941
C
C DATE    : 25/06/91
C
C UPDATE  : 27/04/92 HP SUMMERS  - ADDED DEFAULT YEAR FOR DATA IF
C                                  REQUESTED YEAR DOES NOT EXIST.
C                                  (ADDED 'YEARDF') IN DHDATA
C
C VERSION 1.1 DATE: 28-10-97
C RICHARD MARTIN
C PUT UNDER SCCS CONTROL (ADAS412).
C
C-----------------------------------------------------------------------
C
C
      CHARACTER*2         YEAR,        YEARDF
      INTEGER             IFAIL,       ITMAX,       IZ0
      REAL*8              DDENS(NTDIM),             DDENSH(NTDIM)
      REAL*8              DTEV(NTDIM), FABUND(NTDIM,NZDIM)
```

5.99 lh404rr: Subroutine lh404rr from library adas4xx

```
      SUBROUTINE LH404RR(DATE  , USER  ,
     &                   IZ0   , IZL   , IZH   , CHPRFIX ,
     &                   MAXT  , MAXD  , TEK   , DENSA   ,
     &                   DSNIN , DSNO  , LDTYP , YEAR    , OPEN17
     &                  )
C-----------------------------------------------------------------------
C
C ************ FORTRAN 77 PROGRAM: LH404RR *****************************
C
C   VERSION 1.0
C
C   PURPOSE:
C           TO FETCH DATA FROM MASTER CONDENSED PARENT/METASTABLE
C           RESOLVED COLLISIONAL DIELECTRONIC FILES AND PREPARE
C           RESOLVED ISONUCLEAR (ADF11) MASTER FILES.
C
C           BASED UPON LH404RU
C
C   PROGRAM:
C
C   PARAMETER : (I*4)  NUTMAX - OUTPUT  ELEMENT MASTER FILE
C                               MAXIMUM NUMBER OF TEMPERATURES
C   PARAMETER : (I*4)  NUDMAX - OUTPUT  ELEMENT MASTER FILE
C                               MAXIMUM NUMBER OF DENSITIES
C   PARAMETER : (I*4)  NUZMAX - OUTPUT  ELEMENT MASTER FILE
C                               MAXIMUM NUMBER OF CHARGE STATES
C   PARAMETER : (I*4)  NUMMAX - METASTABLE FRACTIONS
C                               MAXIMUM NUMBER OF METASTABLES
C
C       (R*8)  DENSA()        - OUTPUT ELEMENT MASTER FILE
C                               SET OF MAXD DENSITIES
C       (R*8)  TEK()          - OUTPUT ELEMENT MASTER FILE
C                               SET OF MAXT TEMPERATURES
C       (R*8)  DENSL()        - OUTPUT ELEMENT MASTER FILE
C                               SET OF MAXD DENSITIES IN LOGARITHM
C       (R*8)  TEVL()         - OUTPUT ELEMENT MASTER FILE
C                               SET OF MAXT TEMPERATURES IN LOGARITHM
C
C
C     ROUTINES:
C     ---------
C         XUFLOW   - VS FORTRAN UNDERFLOW EXCEPTION HANDLER
C         CL3270   - JET-SPECIFIC CLEAR SCREEN ROUTINE
C         XXDATE   - ADAS - GATHER CURRENT DATE
C         XXOPEN   - ADAS - OPEN FILE
C         XXSLEN   - ADAS - GET FIRST AND LAST CHAR. POS. IN A STRING
C         DMGUID   - JET-SPECIFIC - GATHER USERS ID
C         FILEINF  - VS FORTRAN FILE INFORMATION ROUTINE
C         CNV404A  - READ ACD,SCD,CCD,PRB,PRC,QCD,XCD RESOLVED
C                    ADF10 FILES AND WRITE RESOLVED ADF11 FILES
C         CNV404B  - READ PLT,PLS RESOLVED ADF10 FILES
C                    AND WRITE RESOLVED ADF11 FILES
C
C-----------------------------------------------------------------------
C AUTHOR:  LORNE D. HORTON
C          ROOM K1/1/58, JET JOINT UNDERTAKING
C
C   DATE:   5TH AUGUST 1996
C
```

```
C-----------------------------------------------------------------------
C UNIX-IDL PORT:
C
C VERSION: 1.1 DATE: 11-11-96
C MODIFIED: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C - FIRST CONVERTED
C               - MADE A SUBROUTINE OF ADAS404
C
C VERSION:  1.2                         DATE: 4/11/99
C MODIFIED: Martin O'Mullane
C           - Add real name of producer.
C           - Modify comments.
C           - Write the metastable line with new D4WMET subroutine.
C           - Increase size of DSNOUT to 80 from 35.
C
C VERSION :  1.2
C DATE    : 23-02-2006
C MODIFIED: Martin O'Mullane
C           - Use the same extrapolation as unresolved case. ie
C             do not bypass options in the dxspl<n>.for routines.
C
C VERSION :  1.3
C DATE    : 04-01-2007
C MODIFIED: Martin O'Mullane
C           - F-like and Ne-like stages have 2 (not 1) metastables.
C             Change NGRD vector.
C           - Use 11605.4 as K/eV conversion (same as xxtcon).
C
C-----------------------------------------------------------------------

      CHARACTER*2         CHPRFIX
      CHARACTER*8         DATE
      CHARACTER*80        DSNIN(50,10),             DSNO(10)
      CHARACTER*30        USER
      CHARACTER*2         YEAR
      INTEGER             IZ0,          IZH,        IZL,          MAXD
      INTEGER             MAXT
      LOGICAL             LDTYP(9),     OPEN17
      REAL*8              DENSA(NUDMAX),            TEK(NUTMAX)
```

5.100 lh404ru: Subroutine lh404ru from library adas4xx

```
       SUBROUTINE LH404RU(DATE  , USER ,
     &                    IZ0   , IZL  , IZH   , CHPRFIX ,
     &                    MAXT  , MAXD , TEK   , DENSA   ,
     &                    DSNIN , DSNO , LDTYP , YEAR    , OPEN17
     &                   )
C-----------------------------------------------------------------------
C
C ************ FORTRAN 77 PROGRAM: LH404RU *****************************
C
C   VERSION 1.0
C
C   PURPOSE:  TO FETCH DATA FROM MASTER CONDENSED PARENT/METASTABLE
C             RESOLVED COLLISIONAL DIELECTRONIC FILES, BUNDLE THEM,
C             AND PREPARE UNRESOLVED ISONUCLEAR (ADF11) MASTER FILES.
C
C             ALSO CHECK FOR AN EQUIVALENTLY NAMED ADF15 FILES
C
C             AND BUNDLE RESOLVED DATA BLOCKS INTO UNRESOLVED ONES.
C
C             LOOSELY BASED UPON WJD404R
C
C
C   PROGRAM:
C
C   PARAMETER : (I*4)  NUTMAX - OUTPUT  ELEMENT MASTER FILE
C                               MAXIMUM NUMBER OF TEMPERATURES
C   PARAMETER : (I*4)  NUDMAX - OUTPUT  ELEMENT MASTER FILE
C                               MAXIMUM NUMBER OF DENSITIES
C   PARAMETER : (I*4)  NUZMAX - OUTPUT  ELEMENT MASTER FILE
C                               MAXIMUM NUMBER OF CHARGE STATES
C   PARAMETER : (I*4)  NUMMAX - METASTABLE FRACTIONS
C                               MAXIMUM NUMBER OF METASTABLES
C
C       (R*8)  DENSA()        - OUTPUT ELEMENT MASTER FILE
C                               SET OF MAXD DENSITIES
C       (R*8)  TEK()          - OUTPUT ELEMENT MASTER FILE
C                               SET OF MAXT TEMPERATURES
C       (R*8)  DENSL()        - OUTPUT ELEMENT MASTER FILE
C                               SET OF MAXD DENSITIES IN LOGARITHM
C       (R*8)  TEVL()         - OUTPUT ELEMENT MASTER FILE
C                               SET OF MAXT TEMPERATURES IN LOGARITHM
C
C
C     ROUTINES:
C     ---------
C         XUFLOW   - VS FORTRAN UNDERFLOW EXCEPTION HANDLER
C         CL3270   - JET-SPECIFIC CLEAR SCREEN ROUTINE
C         XXDATE   - ADAS - GATHER CURRENT DATE
C         XXOPEN   - ADAS - OPEN FILE
C         XXSLEN   - ADAS - GET FIRST AND LAST CHAR. POS. IN A STRING
C         DMGUID   - JET-SPECIFIC - GATHER USERS ID
C         METRD    - READ MET FILES AND SPLINE ONTO TEMP/DENS ARRAY
C         FILEINF  - VS FORTRAN FILE INFORMATION ROUTINE
C         BND404A  - READ ACD,SCD,CCD,PRB,PRC RESOLVED ADF10 FILES
C                    AND BUNDLE THEM INTO UNRESOLVED ADF11 FILES
C         BND404B  - READ PLT,PLS RESOLVED ADF10 FILES
C                    AND BUNDLE THEM INTO UNRESOLVED ADF11 FILES
C         BND404C  - READ ADF10 FILES AND BUNDLE THE RESOLVED
C                    BLOCKS INTO UNRESOLVED BLOCKS TO BE TACKED
```

```
C                    ONTO THE BOTTOM OF THE FILE
C
C-----------------------------------------------------------------------
C AUTHOR:  LORNE D. HORTON
C          ROOM K1/1/58, JET JOINT UNDERTAKING
C
C   DATE:  21ST FEBRUARY 1996
C
C-----------------------------------------------------------------------
C
C VERSION: 1.2
C DATE: 20-10-97
C MODIFIED: LORNE HORTON
C - MODIFIED CALL TO BND404A and BND404B.
C
C VERSION:  1.3
C DATE: 4/11/99
C MODIFIED: Martin O'Mullane
C           - Add real name of producer.
C           - Modify comments.
C
C VERSION :  1.4
C DATE    : 04-01-2007
C MODIFIED: Martin O'Mullane
C           - F-like and Ne-like stages have 2 (not 1) metastables.
C             Change NGRD vector.
C           - Use 11605.4 as K/eV conversion (same as xxtcon).
C
C VERSION :  1.5
C DATE    : 17-05-2007
C MODIFIED: Allan Whiteford
C           - Updated comments as part of subroutine documentation
C             procedure.
C
C-----------------------------------------------------------------------

      CHARACTER*2         CHPRFIX
      CHARACTER*8         DATE
      CHARACTER*80        DSNIN(50,10),            DSNO(10)
      CHARACTER*30        USER
      CHARACTER*2         YEAR
      INTEGER             IZ0,         IZH,        IZL,         MAXD
      INTEGER             MAXT
      LOGICAL             LDTYP(9),    OPEN17
      REAL*8              DENSA(NUDMAX),           TEK(NUTMAX)
```

5.101 lmdif1_all: Subroutine lmdif1_all from library adas4xx

```
      subroutine lmdif1(fcn,m,n,x,fvec,tol,info,iwa,wa,lwa)
C-----------------------------------------------------------------------
c     subroutine lmdif1
c
c     PURPOSE: minimize the sum of the squares of m nonlinear
c     functions in n variables by a modification of the
c     levenberg-marquardt algorithm.
c
c     this is done by using the more general least-squares solver
c     lmdif. the user must provide a subroutine which calculates
c     the functions. the jacobian is then calculated by a
c     forward-difference approximation.
c
c     the subroutine statement is
c
c       subroutine lmdif1(fcn,m,n,x,fvec,tol,info,iwa,wa,lwa)
c
c     where
c
c       fcn is the name of the user-supplied subroutine which
c         calculates the functions. fcn must be declared
c         in an external statement in the user calling
c         program, and should be written as follows.
c
c         subroutine fcn(m,n,x,fvec,iflag)
c         integer m,n,iflag
c         double precision x(n),fvec(m)
c         ----------
c         calculate the functions at x and
c         return this vector in fvec.
c         ----------
c         return
c         end
c
c         the value of iflag should not be changed by fcn unless
c         the user wants to terminate execution of lmdif1.
c         in this case set iflag to a negative integer.
c
c       m is a positive integer input variable set to the number
c         of functions.
c
c       n is a positive integer input variable set to the number
c         of variables. n must not exceed m.
c
c       x is an array of length n. on input x must contain
c         an initial estimate of the solution vector. on output x
c         contains the final estimate of the solution vector.
c
c       fvec is an output array of length m which contains
c         the functions evaluated at the output x.
c
c       tol is a nonnegative input variable. termination occurs
c         when the algorithm estimates either that the relative
c         error in the sum of squares is at most tol or that
c         the relative error between x and the solution is at
c         most tol.
c
c       info is an integer output variable. if the user has
```

```
c         terminated execution, info is set to the (negative)
c         value of iflag. see description of fcn. otherwise,
c         info is set as follows.
c
c         info = 0  improper input parameters.
c
c         info = 1  algorithm estimates that the relative error
c                   in the sum of squares is at most tol.
c
c         info = 2  algorithm estimates that the relative error
c                   between x and the solution is at most tol.
c
c         info = 3  conditions for info = 1 and info = 2 both hold.
c
c         info = 4  fvec is orthogonal to the columns of the
c                   jacobian to machine precision.
c
c         info = 5  number of calls to fcn has reached or
c                   exceeded 200*(n+1).
c
c         info = 6  tol is too small. no further reduction in
c                   the sum of squares is possible.
c
c         info = 7  tol is too small. no further improvement in
c                   the approximate solution x is possible.
c
c       iwa is an integer work array of length n.
c
c       wa is a work array of length lwa.
c
c       lwa is a positive integer input variable not less than
c         m*n+5*n+m.
c
c     subprograms called
c
c       user-supplied ...... fcn
c
c       minpack-supplied ... lmdif
c
c     argonne national laboratory. minpack project. march 1980.
c     burton s. garbow, kenneth e. hillstrom, jorge j. more
c
C PUT INTO ADAS BY:
C    WILLIAM OSBORN, TESSELLA SUPPORT SERVICES PLC.
C
C DATE:    25TH APRIL 1996
C
C VERSION: 1.1 DATE: 25-04-96
C MODIFIED: WILLIAM OSBORN
C     - FOUND AT WWW.NETLIB.ORG/MINPACK/ .
C             REPLACES NAG ROUTINE E04FDF .
C
C VERSION: 1.2 DATE: 21-05-96
C MODIFIED: WILLIAM OSBORN
C           - CHANGED TOLERANCE CALCULATION IN ORDER TO USE SMALLER
C             VALUES
C
C VERSION: 1.3 DATE: 22-09-99
C MODIFIED: RICHARD MARTIN
C  - RENAMED FROM lmdif1_all.f to lmdif1_all.for
C
```

```
C VERSION: 1.4 DATE: 16-05-07
C MODIFIED: Allan Whiteford
C     - Updated comments as part of subroutine documentation
C             procedure.
C
C-----------------------------------------------------------------------
      DOUBLE PRECISION    FVEC(M),     TOL,          WA(LWA),     X(N)
      INTEGER             INFO,        IWA(N),       LWA,         M
      INTEGER             N
      DOUBLE PRECISION    X(N)
      INTEGER             N
      DOUBLE PRECISION    EPSFCN,      FJAC(LDFJAC,N),           FVEC(M)
      DOUBLE PRECISION    WA(M),       X(N)
      INTEGER             IFLAG,       LDFJAC,       M,           N
      DOUBLE PRECISION    DIAG(N),     EPSFCN,       FACTOR
      DOUBLE PRECISION    FJAC(LDFJAC,N),            FTOL,        FVEC(M)
      DOUBLE PRECISION    GTOL,        QTF(N),       WA1(N),      WA2(N)
      DOUBLE PRECISION    WA3(N),      WA4(M),       X(N),        XTOL
      INTEGER             INFO,        IPVT(N),      LDFJAC,      M
      INTEGER             MAXFEV,      MODE,         N,           NFEV
      INTEGER             NPRINT
      DOUBLE PRECISION    DELTA,       DIAG(N),      PAR,         QTB(N)
      DOUBLE PRECISION    R(LDR,N),    SDIAG(N),     WA1(N),      WA2(N)
      DOUBLE PRECISION    X(N)
      INTEGER             IPVT(N),     LDR,          N
      DOUBLE PRECISION    A(LDA,N),    ACNORM(N),    RDIAG(N),    WA(N)
      INTEGER             IPVT(LIPVT), LDA,          LIPVT,       M
      INTEGER             N
      LOGICAL             PIVOT
      DOUBLE PRECISION    DIAG(N),     QTB(N),       R(LDR,N)
      DOUBLE PRECISION    SDIAG(N),    WA(N),        X(N)
      INTEGER             IPVT(N),     LDR,          N
```

5.102 lsfun1: Subroutine lsfun1 from library adas4xx

```
      subroutine lsfun1(npt,nlf,pnlfa,fvecc,iflag)
      implicit real*8(a-h,o-z)

C-----------------------------------------------------------------------
C  PURPOSE : Program to evaluate functionals for least squares to line
C            power in Gaunt factor approximation.
C
C  COMMENT : Specific for application to program 'ADAS.FORT(POWERFIT)'
C            Necessary for input to NAG routine E04FDF
C
C
C  AUTHOR  : H.P. Summers, JET
C  DATE    : 25-2-91
C
C UNIX-IDL PORT:
C    WILLIAM OSBORN, TESSELLA SUPPORT SERVICES PLC.
C
C DATE:    17TH APRIL 1996
C
C VERSION: 1.1 DATE: 17-04-96
C MODIFIED: WILLIAM OSBORN
C     - FIRST VERSION
C
C VERSION: 1.2 DATE: 25-04-96
C MODIFIED: WILLIAM OSBORN
C     - ADDED IFLAG FOR CALLS BY LMDIF1 RATHER THAN E04FDF
C
C-----------------------------------------------------------------------

      INTEGER             IFLAG,       NLF,         NPT
      REAL*8              FVECC(NPT),  PNLFA(NLF)
```

5.103 metrd: Subroutine metrd from library adas4xx

```
       SUBROUTINE METRD(NUTMAX , NUDMAX , NUZMAX , NUMMAX ,
     &                  MAXT   , MAXD   , DSNIN  ,
     &                  IZL    , IZH    , IZ0    ,
     &                  TEK    , DENSA  ,
     &                  METFRC ,
     &                  NGRD   ,
     &                  IST2   , IST5 )
C
C-----------------------------------------------------------------------
C
C ************ FORTRAN 77 SUBROUTINE METRD *****************************
C
C   VERSION 1.0
C
C   PURPOSE:
C           TO FETCH DATA FROM ADF10 'MET' FILES AND SPLINE ONTO
C           THE REQUESTED TEMPERATURE/DENSITY GRID.
C
C           CALLING PROGRAM LH404RU
C
C   SUBROUTINE:
C
C   INPUT : (I*4)  NUTMAX - OUTPUT ELEMENT MASTER FILE
C                           MAXIMUM NUMBER OF TEMPERATURES
C   INPUT : (I*4)  NUDMAX - OUTPUT ELEMENT MASTER FILE
C                           MAXIMUM NUMBER OF DENSITIES
C   INPUT : (I*4)  NUZMAX - OUTPUT ELEMENT MASTER FILE
C                           MAXIMUM NUMBER OF CHARGE STATES
C   INPUT : (I*4)  NUMMAX - OUTPUT ELEMENT MASTER FILE
C                           MAXIMUM NUMBER OF METASTABLES
C   INPUT : (I*4)  MAXT   - OUTPUT ELEMENT MASTER FILE
C                           ACTUAL NUMBER OF TEMPERATURES
C   INPUT : (I*4)  MAXD   - OUTPUT ELEMENT MASTER FILE
C                           ACTUAL NUMBER OF DENSITIES
C   INPUT : (C*80) DSNIN(,) - NAMES OF MASTER CONDENSED FILES
C                             TO BE OPENED
C   INPUT : (I*4)  IZL    - LOWEST ION CHARGE TO READ
C   INPUT : (I*4)  IZH    - HIGHEST ION CHARGE TO READ
C                           ACTUALLY READ ONE MORE IF IZH<IZ0
C   INPUT : (I*4)  IZ0    - NUCLEAR CHARGE TO READ
C   INPUT : (R*8)  DENSA()- OUTPUT ELEMENT MASTER FILE
C                           SET OF MAXD DENSITIES
C   INPUT : (R*8)  TEK()  - OUTPUT ELEMENT MASTER FILE
C                           SET OF MAXT TEMPERATURES
C   OUTPUT: (R*8)  METFRC(,,,) - METASTABLE POPULATION FRACTIONS,
C                           SPLINED ONTO THE OUTPUT TEMPERATURES
C                           AND DENSITIES
C                           1ST DIMENSION - DENSITY INDEX
C                           2ND DIMENSION - TEMPERATURE INDEX
C                           3RD DIMENSION - CHARGE STATE INDEX
C                           4TH DIMENSION - METASTABLE INDEX
C   INPUT : (I*4)  NGRD() - NUMBER OF GROUND STATES OF THE FIRST
C                           50 ISOELECTRONIC SEQUENCES
C   INPUT : (I*4)  IST2   - UNIT NUMBER FOR OUTPUT INFORMATION
C                           AND ERROR MESSAGES
C   INPUT : (I*4)  IST5   - UNIT NUMBER FOR READING MASTER CONDENSED
C                           FILE
C
C   PARAMETER : (I*4)  NTDMAX - SIZE OF LOCAL WORKING SPACE
```

```
C                                    (MUST BE GREATER THAN NUTMAX & NUDMAX)
C   PARAMETER : (I*4)  NDZ1V  - MASTER CONDENSED FILE
C                                  MAXIMUM NUMBER OF CHARGE STATES
C   PARAMETER : (I*4)  NDTIN  - MASTER CONDENSED FILE
C                                  MAXIMUM NUMBER OF TEMPERATURES
C   PARAMETER : (I*4)  NDDEN  - MASTER CONDENSED FILE
C                                  MAXIMUM NUMBER OF DENSITIES
C   PARAMETER : (I*4)  NDMET  - MASTER CONDENSED FILE
C                                  MAXIMUM NUMBER OF METASTABLES
C
C         : (R*8)  DENSR()    - INPUT MASTER CONDENSED FILE
C                                  SET OF IDE REDUCED DENSITIES
C         : (R*8)  TR()       - INPUT MASTER CONDENSED FILE
C                                  SET OF ITE REDUCED TEMPERATURES
C         : (R*8)  ZIPT()     - INPUT MASTER CONDENSED FILE
C                                  SET OF IZE RECOMBINING ION CHARGES
C         : (R*8)  AIPTM(,,)  - INPUT MASTER CONDENSED FILE
C                                  RATIO OF METASTABLE TO GROUND POP.
C                                  1ST DIMENSION - DENSITY INDEX
C                                  2ND DIMENSION - TEMPERATURE INDEX
C                                  3RD DIMENSION - CHARGE STATE INDEX
C                                  4TH DIMENSION - METASTABLE INDEX
C         : (R*8)  EIA()      - INPUT MASTER CONDENSED FILE
C                                  SET OF IONISATION POTENTIALS (CM-1)
C
C         : (R*8)  ATTY(,)    - WORK SPACE FOR INTERPOLATION
C                               - STORES LOG10(INTERPOLATED VALUES)
C                                  1ST DIMENSION - TEMPERATURE
C                                  2ND DIMENSION - DENSITY
C         : (R*8)  ARRAY(,)   - STORES LOG10(INTERPOLATED VALUES)
C                                  1ST DIMENSION - TEMPERATURE
C                                  2ND DIMENSION - DENSITY
C
C     ROUTINES:
C     ---------
C         XXOPEN   -
C         XXTERM   -
C         XXIN80   -  FETCH DATA FROM MASTER CONDENSED FILE
C         D4SPLN   -  INTERPOLATE CONDENSED MASTER FILE
C                     UPDATED VERSION OF D1SPLN
C
C-----------------------------------------
C AUTHOR:  LORNE D. HORTON
C          ROOM K1/1/58, JET JOINT UNDERTAKING
C
C   DATE:  21ST FEBRUARY 1996
C-----------------------------------------------------------------------
C UNIX-IDL PORT:
C
C VERSION: 1.1 DATE: 11-11-96
C MODIFIED: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C     - FIRST CONVERTED
C
C VERSION: 1.2 DATE: 20-11-96
C MODIFIED: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C     - REMOVED DIAGNOSTIC WRITE STATEMENTS
C
C VERSION: 1.3 DATE: 20-10-97
C MODIFIED: LORNE HORTON
C     - REMOVED WHITE SPACE FROM DATA FILENAME.
C
```

```
C-----------------------------------------------------------------------
C
      CHARACTER*80        DSNIN(50,10)
      INTEGER             IST2,         IST5,         IZ0,          IZH
      INTEGER             IZL,          MAXD,         MAXT
      INTEGER             NGRD(50),     NUDMAX,       NUMMAX,       NUTMAX
      INTEGER             NUZMAX
      REAL*8              DENSA(NUDMAX)
      REAL*8              METFRC(NUDMAX,NUTMAX,NUZMAX,NUMMAX)
      REAL*8              TEK(NUTMAX)
```

5.104 nvgoel: Subroutine nvgoel from library adas4xx

```
      SUBROUTINE NVGOEL(MAXT,TEA,GA,GA0,GAREST,Z1,N0,V0,PHFRAC,NCUT)
      IMPLICIT REAL*8(A-H,O-Z)
C-----------------------------------------------------------------------
C  PURPOSE: EVALUATE TOTAL RADIATIVE RECOMBINATION RATE COEFFICIENTS
C  AT ZERO DENSITY USING THE VON GOELER TYPE FORMULA WITH MODIFIED
C  CAPTURE TO THE LOWEST ACCESSIBLE PRINCIPAL QUANTUM SHELL.
C
C  PHFRAC GIVES THE PROPORTION OF THE LOWEST LEVEL CAPTURE ALLOWED
C  BASED ON THE AVAILABLE PHASE SPACE OF OCCUPIED SHELLS ARGUMENTS.
C
C  MODIFICATION OF VGOEL TO EXTEND ARRAYS AND INCLUDE NCUT
C
C  ********** H.P. SUMMERS, JET          24 JUNE 1987  ***************
C  ***********                     MOD.  24 AUG  1989  ***************
C  INPUT
C      MAXT=NUMBER OF TEMPERATURES
C      TEA(I)=ELECTRON TEMPERATURES (K)
C      Z1=RECOMBINING ION CHARGE
C      N0=LOWEST ACCESSIBLE N-SHELL BY RECOMBINATION
C      V0=EFFECTIVE PRINCIPAL QUANTUM NUMBER OF LOWEST ACCESSIBLE SHELL
C      PHFRAC=PHASE SPACE OCCUPATION FACTOR FOR LOWEST ACCESSIBLE SHELL
C      NCUT=CUT-OFF OF MAXIMUM NUMBER OF N-SHELLS
C  OUTPUT
C      GA(I)=TOTAL RADIATIVE RECOMBINATION COEFFICIENT (CM+3 SEC-1)
C      GA0(I)=GROUND SHELL  RECOMBINATION COEFFICIENT
C      GAREST(I)=RECOMBINATION COEFFICIENT TO ALL SHELLS EXCLUDING
C      THE GROUND SHELL.
C
C
C UNIX-IDL PORT:
C    WILLIAM OSBORN, TESSELLA SUPPORT SERVICES PLC.
C
C DATE:    26TH MARCH 1996
C
C VERSION: 1.1 DATE: 26-03-96
C MODIFIED: WILLIAM OSBORN
C     - FIRST VERSION. ERRSET COMMENTED OUT
C
C VERSION: 1.2 DATE: 16-05-07
C MODIFIED: Allan Whiteford
C     - Updated comments as part of subroutine documentation
C             procedure.
C-----------------------------------------------------------------------
      INTEGER             MAXT,        N0,          NCUT
      REAL*8              GA(100),     GA0(100),    GAREST(100), PHFRAC
      REAL*8              TEA(100),    V0,          Z1
```

5.105 rbchid: Subroutine rbchid from library adas4xx

```
       FUNCTION RBCHID(Z,XI,ZETA,TE)

       IMPLICIT REAL*8(A-H,O-Z)

C  PURPOSE: EVALUATES A SHELL CONTRIBUTION TO THE IONISATION RATE
C  COEFFICIENT IN THE BURGESS-CHIDICHIMO APPROXIMATION
C
C  MNRAS(1983)203,1269.
C
C      Z=TARGET ION CHARGE NUMBER
C      XI=EFFECTIVE IONISATION POTENTIAL FOR SHELL (RYD)
C      ZETA=EFFECTIVE NUMBER OF EQUIVALENT ELECTRONS IN SHELL
C      TE=ELECTRON TEMPERATURE (K)
C UNIX-IDL PORT:
C    WILLIAM OSBORN, TESSELLA SUPPORT SERVICES PLC.
C
C DATE:    22ND APRIL 1996
C
C VERSION: 1.1 DATE: 22-04-96
C MODIFIED: WILLIAM OSBORN
C     - FIRST VERSION. NO CHANGES TO IBM CODE.
C
C VERSION: 1.2 DATE: 16-05-07
C MODIFIED: Allan Whiteford
C     - Updated comments as part of subroutine documentation
C             procedure.
C
C-----------------------------------------------------------------------

       RBCHID=0.0D0
       C=2.3D0
       BETA=0.25D0*(DSQRT((100.0D0*Z+91.0)/(4.0D0*Z+3.0D0))-5.0D0)
       ATE=1.5789D5/TE
       Y=ATE*XI
C      WRITE(6,1001) ATE,XI,Y
C      WRITE(7,1001) ATE,XI,Y
 1001  FORMAT(1H0,' ATE,XI,Y = ',1P,3D12.2)
       IF(Y.GT.150.0D0)GO TO 10
       T1=ZETA*DSQRT(Y)*DEXP(-Y)*EEI(Y)/XI**1.5D0
       P=1.0D0+1.0D0/Y
       W=(DLOG(P))**(BETA/P)
       RBCHID=2.1715D-8*C*T1*W
   10  RETURN
      END
      REAL*8              TE,          XI,          Z,           ZETA
```

5.106 xcoef: Subroutine xcoef from library adas4xx

```
       SUBROUTINE XCOEF ( FILELS , IFAIL  , LFXIST ,
     &                    NION   , MAXT   ,
     &                    INDA   , NIND   , NSPEC  ,
     &                    LPSEL  , LZSEL  , LISEL  , LHSEL  , LRSEL  ,
     &                    TEVA   , TPVA   , THVA   ,
     &                    DENSA  , DENSPA , RATHA  , RATIA  ,
     &                    ZEFF   ,
     &                    COEF   , SPEC
     &                  )
C-----------------------------------------------------------------------
C  ****************** FORTRAN77 SUBROUTINE: IONBAL *********************
C
C  VERSION:  1.1
C
C  CALLING PROGRAM: ADAS412
C
C  PURPOSE:  TO  CALCULATE  COMPLETE SETS OF SPECTRUM LINE EMISSIVITIES
C            FOR THE IONS OF AN ELEMENT
C
C            PROCESSES  CAN  INCLUDE  ELECTRON AND PROTON IMPACT,  SPON-
C            TANEOUS EMISSION,   FREE ELECTRON RECOMBINATION  AND CHARGE
C            EXCHANGE RECOMBINATION DEPENDING ON THE INPUT DATA SET.
C
C            ACCEPTS MULTIPLE INPUT FILES.  DESIGNED FOR USE IN G(T)
C            CALCULATIONS ETC.
C
C
C  DATA:     THE SOURCE DATA ARE SPECIFIC ION EXCITATION FILES STORED AS
C            PARTITIONED DATA SET MEMBERS AS FOLLOWS:-
C
C                    'JETSHP.<SE>LIKE.DATA(<MEMBER>)'
C
C            ACCORDING TO ADAS DATA FORMAT ADF04.
C
C
C  INPUT  :(C*60) FILELS()= INPUT COPASE FILE NAMES
C          (L*4)  LFXIST()= .TRUE.  => COPASE FILE FOR THIS ION
C                           .FALSE. => NO COPASE FILE FOR THIS ION
C          (I*4)  NION    = NUMBER OF IONS TO BE COMPUTED
C          (I*4)  MAXT    = NUMBER OF TEMPERATURE/DENSITY PAIRS
C          (L*4)  LPSEL   = .TRUE.  => PROTON DATA TO BE INCLUDED
C                           .FALSE. => PROTON DATA TO BE EXCLUDED
C          (L*4)  LZSEL   = .TRUE.  => SCALE PROTON DATA WITH ZEFF
C                           .FALSE. => DO NOT SCALE PROTON DATA
C          (L*4)  LISEL   = .TRUE.  => IONISATION TO BE INCLUDED
C                           .FALSE. => IONISATION TO BE EXCLUDED
C          (L*4)  LHSEL   = .TRUE.  => CHARGE TRANSFER TO BE INCLUDED
C                           .FALSE. => CHARGE TRANSFER TO BE EXCLUDED
C          (L*4)  LRSEL   = .TRUE.  => RECOMBINATION TO BE INCLUDED
C                           .FALSE. => RECOMBINATION TO BE EXCLUDED
C          (R*8)  TEVA()  = ELECTRON TEMPERATURES (EV)
C          (R*8)  TPVA()  = PROTON TEMPERATURES (EV)
C          (R*8)  THVA()  = NEUTRAL HYDROGEN TEMPERATURES (EV)
C          (R*8)  DENSA() = ELECTRON DENSITIES  (CM-3)
C          (R*8)  DENSPA()= PROTON DENSITIES  (CM-3)
C          (R*8)  RATHA() = RATIO (NEUTRAL H DENSITY/ELECTRON DENSITY)
C          (R*8)  RATIA() = RATIO (N(Z+1)/N(Z) STAGE ABUNDANCES)
C
C  OUTPUT :(I*4)  IFAIL   = 0    SUBROUTINE SUCCESSFUL
```

```
C                            1    SUBROUTINE FAILURE OR WARNING
C          (I*4)  INDA(,) = IDENTIFIER FOR SPECTRUM LINE (10000*IL+IU)
C                           1ST DIMENSION - INDEX OF LINES FOR AN ION
C                           2ND DIMENSION - ION COUNT INDEX
C          (I*4)  NIND()  = NUMBER OF LINES FOR AN ION
C                           1ST DIMENSION - ION COUNT INDEX
C          (I*4)  NSPEC() = NUMBER OF LEVELS FOR AN ION
C                           1ST DIMENSION - ION COUNT INDEX
C          (R*8)  ZEFF    = PLASMA Z EFFECTIVE ( IF 'LZSEL' = .TRUE.)
C                             (IF 'LZSEL' = .FALSE. => 'ZEFF=1.0')
C          (R*8)  COEF(,,)= EMISSIVITY FOR SPECTRUM LINE (10000*J+I)
C                           1ST DIMENSION - INDEX OF LINES FOR AN ION
C                           2ND DIMENSION - TEMPERATURE INDEX
C                           3RD DIMENSION - ION COUNT INDEX
C          (C*51) SPEC(,) = INFORMATION STRING FOR LEVEL
C                           1ST DIMENSION - INDEX OF LEVELS FOR AN ION
C                           2ND DIMENSION - ION COUNT INDEX
C
C
C  PROGRAM:
C          (I*4)  NDLEV   = PARAMETER = MAX. NUMBER OF LEVELS ALLOWED
C          (I*4)  NDTRN   = PARAMETER = MAX. NO. OF TRANSITIONS ALLOWED
C          (I*4)  NDTEM   = PARAMETER = MAX. NO. OF TEMPERATURES ALLOWED
C          (I*4)  NZDIM   = PARAMETER = MAX. NO. OF IONS ALLOWED
C          (I*4)  NDMET   = PARAMETER = MAX. NO. OF METASTABLES ALLOWED
C
C          (I*4)  IUNT10  = PARAMETER = INPUT UNIT FOR COPASE DATA SET
C                                       PASSING FILE.
C          (I*4)  L1      = PARAMETER = 1
C
C          (R*8)  D1      = PARAMETER = 1.0D0
C
C          (I*4)  ICNTE   = NUMBER OF ELECTRON IMPACT TRANSITIONS INPUT
C          (I*4)  ICNTP   = NUMBER OF PROTON IMPACT TRANSITIONS INPUT
C          (I*4)  ICNTR   = NUMBER OF FREE ELECTRON RECOMBINATIONS INPUT
C          (I*4)  ICNTH   = NO. OF CHARGE EXCHANGE RECOMBINATIONS INPUT
C          (I*4)  IL      = INPUT DATA FILE: NUMBER OF ENERGY LEVELS
C          (I*4)  ITRAN   = INPUT DATA FILE: NUMBER OF TRANSITIONS
C          (I*4)  IZ0     = NUCLEAR CHARGE
C          (I*4)  IZ      =  RECOMBINED ION CHARGE
C          (I*4)  IZ1     = RECOMBINING ION CHARGE
C                           (NOTE: IZ1 SHOULD EQUAL IZ+1)
C          (I*4)  MAXLEV  = HIGHEST INDEX LEVEL IN READ TRANSITIONS
C          (I*4)  MAXT    = NO. OF INPUT TEMP/DENS PAIRS ( 1 -> 'NDTEM')
C          (I*4)  NMET    = NUMBER OF METASTABLES (1 <= NMET <= 'NDMET')
C          (I*4)  NORD    = NUMBER OF ORDINARY LEVELS ('IL' - 'NMET')
C          (I*4)  NV      = INPUT DATA FILE: NUMBER OF GAMMA/TEMPERATURE
C                           PAIRS FOR A GIVEN TRANSITION.
C          (I*4)  I       = GENERAL INDEX
C          (I*4)  IT      = TEMPERATURE ARRAY INDEX
C          (I*4)  IS      = ENERGY LEVEL ARRAY INDEX
C
C          (R*8)  TEA()   = INPUT ELECTRON TEMPERATURES (K)
C          (R*8)  TPA()   = INPUT PROTON TEMPERATURES (K)
C          (R*8)  THA()   = INPUT NEUTRAL HYDROGEN TEMPERATURES (K)
C          (R*8)  R8FBCH  = FUNCTION (SEE ROUTINE SECTION BELOW)
C          (R*8)  BWNO    = IONISATION POTENTIAL (CM-1)
C          (R*8)  ZEFFSQ  = 'ZEFF' * 'ZEFF'
C          (R*8)  DMINT   = +1 or -1 DEPENDING ON WHETHER THE  NUMBER OF
C                           ROW   INTERCHANGES   WAS   EVEN   OR    ODD,
C                           RESPECTIVELY, WHEN INVERTING A MATRIX  USING
```

```
C                          'XXMINV'.
C
C          (L*4)  LSOLVE  = .TRUE.  => SOLVE LINEAR EQUATION USING
C                                      'XXMINV'.
C                          .FALSE. =>DO NOT SOLVE LINEAR EQUATION USING
C                                    'XXMINV' - INVERT MATRIX ONLY.
C          (L*4)  OPEN10  = .TRUE.  => FILE ALLOCATED TO UNIT 10.
C                         = .FALSE. => NO FILE ALLOCATED TO UNIT 10.
C          (C*3)  TITLED  = ELEMENT SYMBOL.
C          (C*8)  DATE    = CURRENT DATE AS 'DD/MM/YY'
C          (C*60) DSNINC  = INPUT COPASE DATA SET NAME (MVS DSN)
C          (C*51) CLINE   = LEVEL SPECIFICATION LINE
C
C          (I*4)  IA()    = ENERGY LEVEL INDEX NUMBER
C          (I*4)  ILA()   = QUANTUM NUMBER (L) FOR LEVEL 'IA()'
C          (I*4)  ISA()   = MULTIPLICITY FOR LEVEL 'IA()'
C                           NOTE: (ISA-1)/2 = QUANTUM NUMBER (S)
C          (I*4)  IMETR() = INDEX OF METASTABLE IN COMPLETE LEVEL LIST
C                           (ARRAY SIZE = 'NDMET' )
C          (I*4)  IORDR() = INDEX OF ORDINARY EXCITED LEVELS IN COMPLETE
C                           LEVEL LIST.
C          (I*4)  I1A()   = TRANSITION:
C                            LOWER ENERGY LEVEL INDEX (CASE ' ' & 'P')
C                            NOT USED                 (CASE 'H' & 'R')
C          (I*4)  I2A()   = TRANSITION:
C                            UPPER ENERGY LEVEL INDEX (CASE ' ' & 'P')
C                            CAPTURING    LEVEL INDEX (CASE 'H' & 'R')
C          (I*4)  IETRN() = ELECTRON IMPACT TRANSITION:
C                           INDEX VALUES IN MAIN TRANSITION ARRAYS WHICH
C                           REPRESENT ELECTRON IMPACT TRANSITIONS.
C          (I*4)  IPTRN() = PROTON IMPACT TRANSITION:
C                           INDEX VALUES IN MAIN TRANSITION ARRAYS WHICH
C                           REPRESENT PROTON IMPACT TRANSITIONS.
C          (I*4)  IRTRN() = FREE ELECTRON RECOMBINATION:
C                           INDEX VALUES IN MAIN TRANSITION ARRAYS WHICH
C                           REPRESENT FREE ELECTRON RECOMBINATIONS.
C          (I*4)  IHTRN() = CHARGE EXCHANGE RECOMBINATION:
C                           INDEX VALUES IN MAIN TRANSITION ARRAYS WHICH
C                           REPRESENT CHARGE EXCHANGE RECOMBINATIONS.
C          (I*4)  IE1A()  = ELECTRON IMPACT TRANSITION:
C                            LOWER ENERGY LEVEL INDEX
C          (I*4)  IE2A()  = ELECTRON IMPACT TRANSITION:
C                            UPPER ENERGY LEVEL INDEX
C          (I*4)  IP1A()  = PROTON IMPACT TRANSITION:
C                            LOWER ENERGY LEVEL INDEX
C          (I*4)  IP2A()  = PROTON IMPACT TRANSITION:
C                            UPPER ENERGY LEVEL INDEX
C
C          (R*8)  PAR(,)  =
C          (R*8)  ER()    = ENERGY RELATIVE TO LEVEL 1 (RYDBERGS)
C                           DIMENSION: LEVEL INDEX
C          (R*8)  XIA()   = ENERGY RELATIVE TO ION. POT. (RYDBERGS)
C                           DIMENSION: LEVEL INDEX
C          (R*8)  AA()    = ELECTRON IMPACT TRANSITION: A-VALUE (SEC-1)
C          (R*8)  AVAL()  = TRANSITION:
C                            A-VALUE (SEC-1)          (CASE ' ')
C                            NEUTRAL BEAM ENERGY      (CASE 'H')
C                            NOT USED                 (CASE 'P' & 'R')
C          (R*8)  SCOM(,) = TRANSITION:
C                            GAMMA VALUES             (CASE ' ' & 'P')
C                            RATE COEFFT. (CM3 SEC-1) (CASE 'H' & 'R')
```

```
C                          1ST DIMENSION - TEMPERATURE 'SCEF()'
C                          2ND DIMENSION - TRANSITION NUMBER
C          (R*8)  SCEF()  = INPUT DATA FILE: ELECTRON TEMPERATURES (K)
C          (R*8)  WA()    = ENERGY RELATIVE TO LEVEL 1 (CM-1)
C                           DIMENSION: LEVEL INDEX
C          (R*8)  XJA()   = QUANTUM NUMBER (J-VALUE) FOR LEVEL 'IA()'
C                           NOTE: (2*XJA)+1 = STATISTICAL WEIGHT
C          (R*8)  RHS()   = USED ONLY IF 'LSOLVE=.TRUE.' WHEN CALLING
C                           THE SUBROUTINE 'XXMINV'. CONTAINS THE SET
C                           OF 'N' LINEAR EQUATIONS TO BE SOLVED.
C                           INPUT  TO   'XXMINV': RIGHT HAND SIDE VECTOR
C                           OUTPUT FROM 'XXMINV': SOLUTION VECTOR
C                           (ACTS ONLY AS A DUMMY IN THIS PROGRAM)
C          (R*8)  CIE()   = IONISATION RATE COEFFICIENT VECTOR FOR
C                           FIXED TEMPERATURE.
C                            DIMENSION: ENERGY LEVEL INDEX
C          (R*8)  VHRED() = CHARGE EXCHANGE RECOMBINATION:
C                           VECTOR  OF RECOMBINATION RATE CONTRIBUTIONS
C                           FOR EACH METASTABLE LEVEL.
C                           (UNITS: SEC-1)
C                           VALUES FOR GIVEN TEMPERATURE AND DENSITY.
C                            DIMENSION: METASTABLE LEVEL INDEX
C          (R*8)  VRRED() = FREE ELECTRON RECOMBINATION:
C                           VECTOR  OF RECOMBINATION RATE CONTRIBUTIONS
C                           FOR EACH METASTABLE LEVEL.
C                           (UNITS: SEC-1)
C                           VALUES FOR GIVEN TEMPERATURE AND DENSITY.
C                            DIMENSION: METASTABLE LEVEL INDEX
C
C          (R*8)  EXCRE(,) = ELECTRON IMPACT TRANSITION:
C                            EXCITATION RATE COEFFS (cm**3/s)
C                            PRE  'BXRATE': UNIT GAMMA VALUES
C                            POST 'BXRATE': TRUE VALUES
C                            1st DIMENSION: TEMPERATURE INDEX ('TOUT')
C                            2nd DIMENSION: TRANSITION INDEX
C          (R*8)  DEXCRE(,)= ELECTRON IMPACT TRANSITION:
C                            DE-EXCITATION RATE COEFFS (cm**3/s)
C                            PRE  'BXRATE': UNIT GAMMA VALUES
C                            POST 'BXRATE': TRUE VALUES
C                            1st DIMENSION: TEMPERATURE INDEX ('TOUT')
C                            2nd DIMENSION: TRANSITION INDEX
C          (R*8)  EXCRP(,) = PROTON IMPACT TRANSITION:
C                            EXCITATION RATE COEFFS (cm**3/s)
C                            PRE  'BXRATE': UNIT GAMMA VALUES
C                            POST 'BXRATE': TRUE VALUES
C                            1st DIMENSION: TEMPERATURE INDEX ('TOUT')
C                            2nd DIMENSION: TRANSITION INDEX
C          (R*8)  DEXCRP(,)= PROTON IMPACT TRANSITION:
C                            DE-EXCITATION RATE COEFFS (cm**3/s)
C                            PRE  'BXRATE': UNIT GAMMA VALUES
C                            POST 'BXRATE': TRUE VALUES
C                            1st DIMENSION: TEMPERATURE INDEX ('TOUT')
C                            2nd DIMENSION: TRANSITION INDEX
C          (R*8)  VECH(,)  = CHARGE-EXCHANGE RECOMBINATION:
C                            SPLINED RECOMBINATION RATE COEFFT. VALUES.
C                            1st DIMENSION: TEMPERATURE INDEX ('TOUT')
C                            2nd DIMENSION: CAPTURING LEVEL INDEX.
C          (R*8)  VECR(,)  = FREE ELECTRON RECOMBINATION:
C                            SPLINED RECOMBINATION RATE COEFFT. VALUES.
C                            1st DIMENSION: TEMPERATURE INDEX ('TOUT')
C                            2nd DIMENSION: CAPTURING LEVEL INDEX.
```

```
C          (R*8)  CRA(,)    = A-VALUE  (sec-1)  MATRIX  COVERING  ALL
C                              TRANSITIONS.
C                              1st DIMENSION: ENERGY LEVEL INDEX
C                              2nd DIMENSION: ENERGY LEVEL INDEX
C                             (NOTE: DIAGONAL   ELEMENTS  REPRESENT   THE
C                                    NEGATIVE  SUM  OF  THEIR  RESPECTIVE
C                                    COLUMNS.)
C          (R*8)  CRCE(,)   = ELECTRON IMPACT TRANSITION:
C                              EXCIT'N/DE-EXCIT'N RATE COEFFT MATRIX
C                              COVERING ALL TRANSITIONS (cm**3/s).
C                              VALUES FOR GIVEN TEMPERATURE & TRANSITION
C                              TYPE.
C                              1st DIMENSION: ENERGY LEVEL INDEX
C                              2nd DIMENSION: ENERGY LEVEL INDEX
C                             (NOTE: DIAGONAL   ELEMENTS  REPRESENT   THE
C                                    NEGATIVE  SUM  OF  THEIR  RESPECTIVE
C                                    COLUMNS.)
C          (R*8)  CRCP(,)   = PROTON IMPACT TRANSITION:
C                              EXCIT'N/DE-EXCIT'N RATE COEFFT MATRIX
C                              COVERING ALL TRANSITIONS (cm**3/s).
C                              VALUES FOR GIVEN TEMPERATURE & TRANSITION
C                              TYPE.
C                              1st DIMENSION: ENERGY LEVEL INDEX
C                              2nd DIMENSION: ENERGY LEVEL INDEX
C                             (NOTE: DIAGONAL   ELEMENTS  REPRESENT   THE
C                                    NEGATIVE  SUM  OF  THEIR  RESPECTIVE
C                                    COLUMNS.)
C          (R*8)  CC(,)     = RATE MATRIX COVERING ALL TRANSITIONS
C                              (UNITS: SEC-1)
C                              VALUES FOR GIVEN TEMPERATURE AND DENSITY.
C                              1st DIMENSION: ENERGY LEVEL INDEX
C                              2nd DIMENSION: ENERGY LEVEL INDEX
C          (R*8)  CMAT(,)   = (INVERTED)   RATE  MATRIX  COVERING   ALL
C                              NON-METASTABLE/ORDINARY EXCITED LEVELS.
C                              (UNITS: SEC)
C                              VALUES FOR GIVEN TEMPERATURE AND DENSITY.
C                              PRE  'XXMINV' : NOT-INVERTED
C                              POST 'XXMINV' : INVERTED
C                              1st DIMENSION: ORDINARY EXCITED LEVEL INDEX
C                              2nd DIMENSION: ORDINARY EXCITED LEVEL INDEX
C          (R*8)  CRED(,)   = MATRIX  OF  TRANSITION   RATES   BETWEEN
C                              METASTABLE LEVELS.
C                              (UNITS: SEC-1)
C                              VALUES FOR GIVEN TEMPERATURE AND DENSITY.
C                              1st DIMENSION: METASTABLE LEVEL INDEX
C                              2nd DIMENSION: METASTABLE LEVEL INDEX
C          (R*8)  CRMAT(,) = INVERTED  METASTABLE  LEVEL   RATE   MATRIX
C                              COVERING ALL TRANSITIONS BETWEEN METASTABLE
C                              LEVELS EXCEPT THOSE INVOLVING LEVEL 1.
C                              VALUES FOR GIVEN TEMPERATURE AND DENSITY.
C                              BEFORE INPUT  TO   XXMINV: NOT INVERTED
C                              AFTER  OUTPUT FROM XXMINV: AS-ABOVE
C                              1st DIMENSION: METASTABLE LEVEL INDEX - 1
C                              2nd DIMENSION: METASTABLE LEVEL INDEX - 1
C
C
C          (R*8) POPAR(,)   = LEVEL POPULATIONS
C                               1st DIMENSION: LEVEL INDEX
C                               2nd DIMENSION: TEMPERATURE INDEX
C          (R*8) STVR(,)    = ORDINARY EXCITED LEVEL:
C                              FREE-ELECTRON RECOMBINATION COEFFICIENTS
```

```
C                              (UNITS* CM**3/SEC-1)
C                              1st DIMENSION: ORDINARY LEVEL INDEX
C                              2nd DIMENSION: TEMPERATURE INDEX
C          (R*8) STVH(,)    = ORDINARY EXCITED LEVEL:
C                              CHARGE-EXCHANGE RECOMBINATION COEFFICIENTS
C                              (UNITS* CM**3/SEC-1)
C                              1st DIMENSION: ORDINARY LEVEL INDEX
C                              2nd DIMENSION: TEMPERATURE INDEX
C          (R*8) STACK(,,) = ORDINARY EXCITED LEVEL POPULAT'N DEPENDENCE
C                            ON METASTABLE LEVEL.
C                              1st DIMENSION: ORDINARY LEVEL INDEX
C                              2nd DIMENSION: METASTABLE INDEX
C                              3rd DIMENSION: TEMPERATURE INDEX
C          (R*8) STVRM(,)   = METASTABLE LEVEL:
C                              FREE-ELECTRON RECOMBINATION COEFFICIENTS
C                              (UNITS* CM**3/SEC-1)
C                              1st DIMENSION: METASTABLE INDEX
C                              2nd DIMENSION: TEMPERATURE INDEX
C          (R*8) STVHM(,)   = METASTABLE LEVEL:
C                              CHARGE-EXCHANGE RECOMBINATION COEFFICIENTS
C                              (UNITS* CM**3/SEC-1)
C                              1st DIMENSION: METASTABLE INDEX
C                              2nd DIMENSION: TEMPERATURE INDEX
C          (R*8) STCKM(,)   = METASTABLE POPULATIONS STACK
C                              1st DIMENSION: METASTABLE INDEX
C                              2nd DIMENSION: TEMPERATURE INDEX
C
C          (C*1)  TCODE() = TRANSITION: DATA TYPE POINTER:
C                           ' ' => Electron Impact   Transition
C                           'P' => Proton   Impact   Transition
C                           'H' => Charge   Exchange Recombination
C                           'R' => Free     Electron Recombination
C          (C*18) CSTRGA()= NOMENCLATURE/CONFIGURATION FOR LEVEL 'IA()'
C          (C*22) STRGA() = LEVEL DESIGNATIONS
C
C          (L*4)  LTRNG(,)= .TRUE.  => TEMPERATURE VALUE WITHIN RANGE
C                                      READ FROM INPUT COPASE DATA SET.
C                         = .FALSE. =>TEMPERATURE VALUE NOT WITHIN RANGE
C                                      READ FROM INPUT COPASE DATA SET.
C                           1st DIMENSION: TEMPERATURE INDEX.
C                           2nd DIMENSION: TEMPERATURE TYPE -
C                                          1) => ELECTRON
C                                          2) => PROTON
C                                          3) => NEUTRAL HYDROGEN
C
C NOTE:
C
C          INPUT/OUTPUT STREAM ALLOCATIONS:
C          --------------------------------
C
C          STREAM 10: INPUT - SPECIFIC ION RATE DATA INPUT FILE FROM
C          ('IUNT10')         DATABASE (SEE DATA SECTION ABOVE).
C
C
C AUTHOR:  HP SUMMERS
C          K1/1/57
C          JET EXT. 4941
C
C DATE:    27/06/91
C
C UPDATE:  12/04/94 - H. P. SUMMERS - RATIONALISING OF DIMENSIONS WITH
```

```
C                                   LATEST ADAS9120 ROUTINES. NOTED
C                                   STACK IS REAL*4
C
C#
C DATE:    mar20-95 - A. C. Lanzafame - conversion to Unix
C          mar21-95 - A. C. Lanzafame - call to XXDATE avoided: redundant
C          mar24-95 -                 - FILELS from C*44 to C*60
C                                     - DSNINC from C*44 to C*60
C          apr27-95 - A. C. Lanzafame - STACK changed to R*8
C
C  ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C      DCSTKC      ADAS      STACK UP TRANSITION RATE BETWEEN METS.
C      DCSTKA      ADAS      STACK UP ORDINARY POP. DEPENDENCE ON MET
C      DCPOPM      ADAS      CALCULATE BASIC MET. LEVEL POPULATIONS.
C      DCPOPO      ADAS      CALCULATE ORDINARY LEVEL POPULATIONS.
C      DCLNORM     ADAS      NORMALISES LINE EMISSIVITY.
C      BXDATA      ADAS      GATHERS RELEVANT DATA FROM INPUT FILE
C      BXTTYP      ADAS      SORT TRANSITIONS INTO TRAN/RECOMB TYPES
C      BXIORD      ADAS      SETS  UP ORDINARY LEVEL INDEX.
C      BXRATE      ADAS      CALCULATES EXC. & DE-EXC. RATE COEFFTS.
C      BXRCOM      ADAS      ESTABLISHES RECOMBINATION RATE COEFFTS.
C      BXMCRA      ADAS      CONSTRUCTS A-VALUE MATRIX.
C      BXMCRC      ADAS      CONSTRUCTS EXC./DE-EXC. RATE COEF MATRIX
C      BXMCCA      ADAS      CONSTRUCTS WHOLE RATE MATRIX.
C      BXMCMA      ADAS      CONSTRUCTS ORDINARY LEVEL RATE MATRIX.
C      BXSTKB      ADAS      STACK UP RECOMB. CONTRIBUTION FOR ORD.
C      BXSTKD      ADAS      STACK UP RECOMB RATE FOR EACH MET. LEVEL
C      BXMPOP      ADAS      CALCULATE METASTABLE LEVEL POPULATIONS.
C      BXSTVM      ADAS      CALCULATE MET. LEVEL RECOMB. COEFFTS.
C      XXERYD      ADAS      CONVERTS ENERGIES FROM W.NO. TO RYDBERGS
C      XXRATE      ADAS      CALCULATES EXC. & DE-EXC. RATE COEFFTS.
C      XXMINV      ADAS      INVERTS MATRIX AND SOLVES EQUATIONS.
C                                FOR UNIT GAMMA VALUE
C      R8FBCH      ADAS      REAL*8 FUNCTION:EVALUATES SHELL CONTRIB.
C                                TO IONISATION RATE  COEFFICIENT  IN  THE
C                                BURGESS-CHIDICHIMO APPROX.
C
C VERSION 1.1 DATE: 28-10-97
C RICHARD MARTIN
C PUT UNDER SCCS CONTROL (ADAS412).
C
C VERSION: 1.2                              DATE: 02-05-2003
C MODIFIED: Martin O'Mullane
C - Use xxdata_04 to read adf04 file. This requires
C                 new arrays some of which are not used in the
C                 population calculation.
C               - bxttyp parameter list extended.
C
C VERSION : 1.3
C DATE    : 04-08-2008
C MODIFIED: Martin O'Mullane
C - Increase number of transitions to 3000.
C
C VERSION : 1.4
C DATE    : 29-08-2008
C MODIFIED: Martin O'Mullane
C - Reduce nzdim from 20 to 1.
C
C-----------------------------------------------------------------------
```

```
CHARACTER*60        FILELS(NZDIM)
CHARACTER*51        SPEC(NDLEV,NZDIM)
INTEGER             IFAIL,       INDA(NDTRN,NZDIM),        MAXT
INTEGER             NIND(NZDIM), NION,        NSPEC(NZDIM)
LOGICAL             LFXIST(NZDIM),            LHSEL,       LISEL
LOGICAL             LPSEL,       LRSEL,       LZSEL
REAL*8              COEF(NDTRN,NDTEM,NZDIM),  DENSA(NDTEM)
REAL*8              DENSPA(NDTEM),            RATHA(NDTEM)
REAL*8              RATIA(NDTEM),             TEVA(NDTEM)
REAL*8              THVA(NDTEM), TPVA(NDTEM), ZEFF
```

6 Subroutine library adas5xx

6.1 e1chkb: Subroutine e1chkb from library adas5xx

```
      SUBROUTINE E1CHKB( IUNIT  , NBSEL  , IBSEL  ,
     &                   IZ0IN  , IZIN   ,
     &                   IZ0    , IZ     ,
     &                   LOPEN  , IRCODE
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: E1CHKB *********************
C
C  PURPOSE: TO CHECK THE SELECTED BLOCK (IBSEL) OF DATA  EXISTS  IN  THE
C           INPUT DATA SET.
C
C           IF SO IT REPRESENTS THE ENTERED VALUES OF
C           'IZ0IN' (NUCLEAR CHARGE OF EMITTING ION)          &
C           'IZIN'  (CHARGE OF EMITTING ION)
C
C           IT ALSO CLOSES THE INPUT DATA SET ALLOCATION IF OPEN.
C
C  CALLING PROGRAM: SSXB
C
C  SUBROUTINE:
C
C  INPUT : (I*4)   IUNIT   = UNIT TO WHICH INPUT DATA SET IS ALLOCATED
C  INPUT : (I*4)   NBSEL   = TOTAL NUMBER OF DATA-BLOCKS READ FROM INPUT
C                            DATA SET.
C  INPUT : (I*4)   IBSEL   = INDEX OF DATA-BLOCK SELECTED FOR ANALYSIS
C
C  INPUT : (I*4)   IZ0IN   = REQUESTED: NUCLEAR CHARGE OF EMITTING ION
C  INPUT : (I*4)   IZIN    = REQUESTED: CHARGE OF EMITTING ION
C
C  INPUT : (I*4)   IZ0     = INPUT FILE: NUCLEAR CHARGE OF EMITTING ION
C  INPUT : (I*4)   IZ      = INPUT FILE: CHARGE OF EMITTING ION
C
C  I/O   : (L*4)   LOPEN   = INPUT : .TRUE.  => INPUT DATA SET OPEN.
C                                    .FALSE. => INPUT DATA SET CLOSED.
C                            OUTPUT: ALWAYS RETURNED AS .FALSE.
C  OUTPUT: (I*4)   IRCODE  = RETURN CODE FROM SUBROUTINE:
C                            0 => NO ERROR DETECTED.
C                            2 => DISCREPANCY BETWEEN REQUESTED CHARGES
C                                 AND THOSE IN INPUT DATA FILE.
C                            3 => SELECTED DATA-BLOCK  OUT OF RANGE  OR
C                                 DOES NOT EXIST.
C
C           (I*4)   I4UNIT  = FUNCTION (SEE ROUTINE SECTION BELOW)
C
CA          (c*80)  DSNAME  = NAME OF DATA FILE INCLUDING PATH.
C
C ROUTINES:
C           ROUTINE    SOURCE    BRIEF DESCRIPTION
C           ------------------------------------------------------------
C           E1FILE     ADAS      OPEN DATA SET FOR SELECTED EMITTER
C           I4UNIT     ADAS      FETCH UNIT NUMBER FOR OUTPUT OF MESSAGES
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/37
C          JET EXT. 2520
C
```

```
C DATE:    03/05/91
C
C UPDATE:  23/04/93 - PE BRIDEN - ADAS91: ADDED I4UNIT FUNCTION TO WRITE
C                                         STATEMENTS FOR SCREEN MESSAGES
C
C UPDATE:  24/05/93 - PE BRIDEN - ADAS91: CHANGED I4UNIT(0)-> I4UNIT(-1)
C UPDATE:  31/10/94 - LALIT JALOTA - INCREASED SIZE DSNAME TO *80 FOR
C UNIX.
C
C-----------------------------------------------------------------------
C
C VERSION: 1.1 DATE: 02-05-95
C MODIFIED: UNKNOWN (SOMEONE FROM TESSELLA SUPPORT SERVICES PLC)
C     - PUT UNDER SCCS CONTROL
C
C VERSION: 1.2 DATE: 20-07-07
C MODIFIED: Allan Whiteford
C     - Small modification to comments to allow for automatic
C             documentation preparation.
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IBSEL,       IRCODE,      IUNIT,       IZ
      INTEGER             IZ0,         IZ0IN,       IZIN,        NBSEL
      LOGICAL             LOPEN
```

6.2 e1data: Subroutine e1data from library adas5xx

```
       SUBROUTINE E1DATA( IUNIT  , DSNAME ,
     &                    NSTORE , NTDIM  , NDDIM  ,
     &                    IZ0    , IZ     , IZ1    , ESYM  ,
     &                    NBSEL  , ISELA  ,
     &                    CWAVEL , CFILE  , CPCODE , CINDM ,
     &                    ITA    , IDA    ,
     &                    TETA   , TEDA   ,
     &                    SXB
     &                  )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: E1DATA *********************
C
C  PURPOSE:  TO  FETCH  DATA  FROM  INPUT IONIZATIONS PER PHOTON FILE
C            FOR A GIVEN EMITTING ION (ELEMENT AND CHARGE).
C            (MEMBER STORED IN IONELEC.DATA - MEMBER PREFIX 'SXB#').
C
C  CALLING PROGRAM: ADAS501/SSXB
C
C  DATA:
C
C           UP TO 'NSTORE' SETS (DATA-BLOCKS) OF DATA MAY BE  READ FROM
C           THE FILE - EACH BLOCK FORMING A COMPLETE SET OF IONIZATIONS
C           PER PHOTON VALUES FOR GIVEN TEMP./DENSITY COMBINATION. EACH
C           DATA-BLOCK  IS  ANALYSED INDEPENDENTLY OF ANY  OTHER  DATA-
C           BLOCK.
C
C           THE UNITS USED IN THE DATA FILE ARE TAKEN AS FOLLOWS:
C
C           TEMPERATURES        : EV
C           DENSITIES           : CM-3
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  IUNIT    = UNIT TO WHICH INPUT FILE IS ALLOCATED.
C  INPUT : (C*80) DSNAME   = NAME OF DATA FILE INCLUDING PATH
C
C  INPUT : (I*4)  NSTORE   = MAXIMUM NUMBER  OF  INPUT DATA-BLOCKS  THAT
C                            CAN BE STORED.
C  INPUT : (I*4)  NTDIM    = MAX NUMBER OF ELECTRON TEMPERATURES ALLOWED
C  INPUT : (I*4)  NDDIM    = MAX NUMBER OF ELECTRON DENSITIES    ALLOWED
C
C  OUTPUT: (I*4)  IZ0      = READ - EMITTING ION - NUCLEAR CHARGE
C  OUTPUT: (I*4)  IZ       = READ - EMITTING ION - CHARGE
C  OUTPUT: (I*4)  IZ1      = READ - EMITTING ION - CHARGE + 1
C  OUTPUT: (C*2)  ESYM     = READ - EMITTING ION - ELEMENT SYMBOL
C
C  OUTPUT: (I*4)  NBSEL    = NUMBER OF DATA-BLOCKS ACCEPTED & READ IN.
C  OUTPUT: (I*4)  ISELA()  = READ - DATA-SET DATA-BLOCK ENTRY INDICES
C                            DIMENSION: DATA-BLOCK INDEX
C
C  OUTPUT: (C*10) CWAVEL() = READ - WAVELENGTH (ANGSTROMS)
C                            DIMENSION: DATA-BLOCK INDEX
C  OUTPUT: (C*8)  CFILE()  = READ - SPECIFIC ION FILE SOURCE
C                            DIMENSION: DATA-BLOCK INDEX
C  OUTPUT: (C*8)  CPCODE() = READ - SPECIFIC ION PROCESSING CODE
C                            DIMENSION: DATA-BLOCK INDEX
C  OUTPUT: (C*2)  CINDM()  = READ - METASTABLE INDEX
C                            DIMENSION: DATA-BLOCK INDEX
```

```
C
C  OUTPUT: (I*4)  ITA()    = READ - NUMBER OF ELECTRON TEMPERATURES
C                            DIMENSION: DATA-BLOCK INDEX
C  OUTPUT: (I*4)  IDA()    = READ - NUMBER OF ELECTRON DENSITIES
C                            DIMENSION: DATA-BLOCK INDEX
C
C  OUTPUT: (R*8)  TETA(,)  = READ - ELECTRON TEMPERATURES (UNITS: eV)
C                            1st DIMENSION: ELECTRON TEMPERATURE INDEX
C                            2nd DIMENSION: DATA-BLOCK INDEX
C  OUTPUT: (R*8)  TEDA(,)  = READ - ELECTRON DENSITIES (UNITS: CM-3)
C                            1st DIMENSION: ELECTRON DENSITY INDEX
C                            2nd DIMENSION: DATA-BLOCK INDEX
C
C  OUTPUT: (R*8)  SXB(,,)   =READ - FULL SET OF IONIZATIONS PER PHOTON
C                                   VALUES.
C                            1st DIMENSION: ELECTRON TEMPERATURE INDEX
C                            2nd DIMENSION: ELECTRON DENSITY INDEX
C                            3rd DIMENSION: DATA-BLOCK INDEX
C
C          (I*4)  I4EIZ0   = FUNCTION - (SEE ROUTINES SECTION BELOW)
C          (I*4)  I4FCTN   = FUNCTION - (SEE ROUTINES SECTION BELOW)
C          (I*4)  I4UNIT   = FUNCTION - (SEE ROUTINE SECTION BELOW)
C          (I*4)  IBLK     = ARRAY INDEX: DATA-BLOCK INDEX
C          (I*4)  ITT      = ARRAY INDEX: ELECTRON TEMPERATURE INDEX
C          (I*4)  ITD      = ARRAY INDEX: ELECTRON DENSITY     INDEX
C          (I*4)  NTNUM    = NUMBER OF ELECTRON TEMPERATURES FOR CURRENT
C                            DATA-BLOCK
C          (I*4)  NDNUM    = NUMBER OF ELECTRON DENSITIES    FOR CURRENT
C                            DATA-BLOCK
C          (I*4)  IABT     = RETURN CODE FROM 'I4FCTN'
C          (I*4)  IPOS1    = GENERAL USE STRING INDEX VARIABLE
C          (I*4)  IPOS2    = GENERAL USE STRING INDEX VARIABLE
C
C          (L*4)  LBEND    = IDENTIFIES WHETHER THE LAST OF THE  INPUT
C                            DATA SUB-BLOCKS HAS BEEN LOCATED.
C                            (.TRUE. => END OF SUB-BLOCKS REACHED)
C
C          (C*1)  CSLASH   = '/' - DELIMITER FOR 'XXHKEY'
C          (C*2)  C2       = GENERAL USE TWO BYTE CHARACTER STRING
C          (C*5)  IONNAM   = EMITTING ION READ FROM DATASET
C          (C*6)  CKEY1    = 'FILMEM' - INPUT BLOCK HEADER KEY
C          (C*4)  CKEY2    = 'CODE  ' - INPUT BLOCK HEADER KEY
C          (C*4)  CKEY3    = 'INDM  ' - INPUT BLOCK HEADER KEY
C          (C*4)  CKEY4    = 'ISEL  ' - INPUT BLOCK HEADER KEY
C          (C*80) C80      = GENERAL USE 80 BYTE  CHARACTER  STRING  FOR
C                            THE INPUT OF DATA-SET RECORDS.
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          XXHKEY     ADAS      OBTAIN KEY/RESPONSE STRINGS FROM TEXT
C          I4EIZ0     ADAS      INTEGER*4 FUNCTION    -
C                               RETURNS Z0 FOR GIVEN ELEMENT SYMBOL
C          I4FCTN     ADAS      INTEGER*4 FUNCTION    -
C                               CONVERT CHARACTER STRING TO INTEGER
C          I4UNIT     ADAS      INTEGER*4 FUNCTION    -
C                               FETCH UNIT NUMBER FOR OUTPUT OF MESSAGES
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/37
C          JET EXT. 4569
```

```
C
C DATE:    30/04/91
C
C UPDATE:  05/12/91 - PE BRIDEN: IONNAM NOW ALLOWED TO OCCUPY EITHER
C                                4 OR 5 SPACES IN THE HEADER.
C
C UPDATE:  23/04/93 - PE BRIDEN - ADAS91: ADDED I4UNIT FUNCTION TO WRITE
C                                         STATEMENTS FOR SCREEN MESSAGES
C
C UPDATE:  24/05/93 - PE BRIDEN - ADAS91: CHANGED I4UNIT(0)-> I4UNIT(-1)
C
C UPDATE:  22/11/94 - L. JALOTA - MODIFIED DSNAME LENGTH FOR UNIX
C
C-----------------------------------------------------------------------
      CHARACTER*8         CFILE(NSTORE)
      CHARACTER*2         CINDM(NSTORE)
      CHARACTER*8         CPCODE(NSTORE)
      CHARACTER*10        CWAVEL(NSTORE)
      CHARACTER*80        DSNAME
      CHARACTER*2         ESYM
      INTEGER             IDA(NSTORE), ISELA(NSTORE)
      INTEGER             ITA(NSTORE), IUNIT,       IZ,          IZ0
      INTEGER             IZ1,         NBSEL,       NDDIM,       NSTORE
      INTEGER             NTDIM
      REAL*8              SXB(NTDIM,NDDIM,NSTORE),  TEDA(NDDIM,NSTORE)
      REAL*8              TETA(NTDIM,NSTORE)
```

6.3 e1file: Subroutine e1file from library adas5xx

```
      SUBROUTINE E1FILE( IUNIT , IZ0 , IZ , IRCODE , DSNAME )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: E1FILE *********************
C
C  PURPOSE: TO OPEN AN IONIZATIONS PER PHOTON DATA SET
C
C           BY DEFAULT AN 'IONELEC' FILE WILL BE USED FOR AN
C           EMITTING ION WITH NUCLEAR CHARGE 'IZ0' AND CHARGE 'IZ'.
C           OR, AN ALTERNATIVE DATA SET CAN BE SPECIFICED.
C           THIS WILL BE CONNECTED TO UNIT 'IUNIT'.
C
CA  DATA FILE OPENED UNDER UNIX : '$ADASCENT/DEFADF/USRGRP/USRTYP/
CA     USRGRP_<EXT><ELEMENT><CHARGE>'
C
C  CALLING PROGRAM: SSXB
C
C  SUBROUTINE:
C
C  INPUT : (I*4)   IUNIT   = UNIT TO WHICH DATA SET WILL BE CONNECTED
C  INPUT : (I*4)   IZ0     = NUCLEAR CHARGE OF EMITTING ION REQUESTED
C  INPUT : (I*4)   IZ      = ION CHARGE OF EMITTING ION REQUESTED
C
C  OUTPUT: (I*4)   IRCODE  = RETURN CODE FROM SUBROUTINE:
C                            0 => DATA SET SUCCESSFULLY CONNECTED
C                            1 => REQUESTED DATA  SET  MEMBER  DOES  NOT
C                                 EXISTS - DATA SET NOT CONNECTED.
C                            6 => INVALID VALUE FOR 'IZ' ENTERED.
C                                 ( 0 <= 'IZ' <= 99 )
C                            9 => REQUESTED DATA  SET  EXISTS BUT CANNOT
C                                 BE OPENED.
CA OUTPUT: (C*80)  DSNAME  = FILE NAME INCLUDING PATH
C
C          (I*4)   IDLEN   = LENGTH, IN BYTES, OF FIXED 'DSNAME' PREFIX
C          (I*4)   LENF1   = FIRST NON-BLANK CHR OF 'DSNAME' GROUP PART
C          (I*4)   LENF2   = LAST  NON-BLANK CHR OF 'DSNAME' GROUP PART
C          (I*4)   LENF3   = FIRST NON-BLANK CHR OF 'DSNAME' TYPE PART
C          (I*4)   LENF4   = LAST  NON-BLANK CHR OF 'DSNAME' TYPE PART
C          (I*4)   LENF5   = FIRST NON-BLANK CHR OF 'DSNAME' USERID PART
C          (I*4)   LENF6   = LAST  NON-BLANK CHR OF 'DSNAME' URESID PART
C          (I*4)   LENF7   = FIRST NON-BLANK CHR OF 'DSNAME' EXTENSION PART
C          (I*4)   LENF8   = LAST  NON-BLANK CHR OF 'DSNAME' EXTENSION PART
C          (I*4)   IZEND   = LAST BYTE WRITTEN TO IN 'CZ'. (= 1 OR 2)
CA         (C*1)   HASH    = '#' if NON-BLANK EXT, ELSE ' '.
C          (C*2)   CZ      = 'IZ' (NO LEADING BLANKS)
C          (C*2)   XFESYM  = FUNCTION - (SEE ROUTINES SECTION BELOW)
C          (C*2)   ESYM    = ELEMENT SYMBOL FOR NUCLEAR CHARGE 'IZ0'
C          (C*3)   USREXT  = ADAS SOURCE DATA FILE EXTENSION
CA         (C*80)  USERID  = ADAS SOURCE DATA USER ID
C          (C*8)   USRGRP  = ADAS SOURCE DATA GROUPNAME
CA         (C*80)  USRTYP  = OPTIONAL SUB-DIRECTORY
CA    (C*6)   DEFADF  = DEFAULT ADF DATA DIRECTORY, I.E. ADF13
C
C          (L*4)   LEXIST  = .TRUE.  => REQUESTED  DATA  SET  EXISTS.
C                            .FALSE. => REQUESTED  DATA  SET  DOES  NOT
C                                       EXIST.
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
```

```
C          ------------------------------------------------------------
C          XXUID      ADAS      FETCHES/SETS ADAS SOURCE DATA USER ID
C          XXSSXB     ADAS      FETCHES/SETS ADAS SOURCE DATA FILENAME
C                               AND FILE EXTENSION
C          XFESYM     ADAS      CHARACTER*2 FUNCTION -
C                               GATHERS ELEMENT SYMBOL FOR NUC. CHARGE
C    XXSLEN     ADAS      FINDS FIRST AND LAST NON-BLANK
C CHARACTERS IN STRING.
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/37
C          JET EXT. 5023
C
C DATE:    01/06/91
C UPDATE:  31/07/91 - PE BRIDEN: ADDED "ACTION='READ'" TO OPEN STATEMENT
C UPDATE:  08/08/91 - PE BRIDEN: ADDED "IRCODE=9" => OPEN ERROR
C UPDATE:  10/03/93 - PE BRIDEN: ADDED CALL TO XXUID AND USERID VARIABLE
C                                - NOW ALLOWS ANY INPUT DATASET USER ID.
C UPDATE:   2/09/93 - HPS      : ADDED CALL TO XXSSXB AND USRGRP, USRTYP
C                                  AND USREXT NAMES
C                                - NOW ALLOWS ANY INPUT DATASET FILENAME
C                                  AND EXTENSION
C UPDATE:  23/11/93 - PEB      : CORRECT ERROR - A '.' HAD MISTAKENLY
C                                BEEN PLACED BEFORE THE MEMBER NAME IN
C                                VARIABLE DSNAME.
C
C UPDATE:  26/10/94 - L. JALOTA: MODIFIED CODE FOR RUNNING UNDER UNIX
C  USING NEW FILENAMING CONVENTION.
C  "ACTION" KEYWORD IN OPEN COMMAND IS IBM
C  SO REMOVED HERE.
C
C UPDATE:   7/11/94 - L. JALOTA: ADDED VARIABLE DEFADF - DEFAULT ADF DIRECTORY
C
C-----------------------------------------------------------------------
C
C VERSION: 1.1 DATE: 02-05-95
C MODIFIED: UNKNOWN (SOMEONE FROM TESSELLA SUPPORT SERVICES PLC)
C     - PUT UNDER SCCS CONTROL
C
C
C VERSION: 1.2 DATE: 19-05-95
C MODIFIED: Hugh Summers
C     - Introduced hash to eliminate # in file if there is
C             no extension part of the file name alter logic to
C             allow usrtyp, usrext to be a single character.
C
C VERSION: 1.3 DATE: 20-07-07
C MODIFIED: Allan Whiteford
C     - Modification to comments to allow for automatic
C             documentation preparation.
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      CHARACTER*80        DSNAME
      INTEGER             IRCODE,      IUNIT,       IZ,          IZ0
```

6.4 e1spln: Subroutine e1spln from library adas5xx

```
      SUBROUTINE E1SPLN( NTDIM  , NDDIM  ,
     &                   ITA    , IDA    , ITVAL   ,
     &                   TETA   , TEDA   , TEVA    , DIN    ,
     &                   SXB    ,          SXBA    ,
     &                                     LTRNG   , LDRNG
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: E1SPLN *********************
C
C  PURPOSE:
C          PERFORMS CUBIC SPLINE ON LOG(TEMPERATURE AND DENSITY)
C          VERSUS LOG(IONIZATIONS PER PHOTON)
C          INPUT DATA FOR A GIVEN WAVELENGTH DATA-BLOCK.
C
C          USING  TWO-WAY SPLINES IT CALCULATES  THE  IONIZATIONS/PHOTON
C          FOR  'ITVAL'  PAIRS OF  ELECTRON TEMPERATURES  AND  DENSITIES
C          FROM THE TWO-DIMENSIONAL TABLE OF TEMPERATURES/DENSITIES READ
C          IN FROM THE INPUT FILE. IF A  VALUE  CANNOT  BE  INTERPOLATED
C          USING SPLINES IT IS EXTRAPOLATED VIA 'XXSPLE'.
C
C  CALLING PROGRAM: ADAS501/SSXB
C
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  NTDIM   = MAX NUMBER OF ELECTRON TEMPERATURES ALLOWED
C  INPUT : (I*4)  NDDIM   = MAX NUMBER OF ELECTRON DENSITIES    ALLOWED
C
C  INPUT : (I*4)  ITA     = INPUT DATA FILE: NUMBER OF ELECTRON TEMPERA-
C                           TURES READ FOR THE DATA-BLOCK BEING ASSESSED
C  INPUT : (I*4)  IDA     = INPUT DATA FILE: NUMBER OF ELECTRON DENSIT-
C                           IES   READ FOR THE DATA-BLOCK BEING ASSESSED
C  INPUT : (I*4)  ITVAL   = NUMBER OF ISPF ENTERED TEMPERATURE/DENSITY
C                           PAIRS  FOR  WHICH  IOINIZATIONS PER PHOTON
C                           ARE REQUIRED FOR TABULAR/GRAPHICAL OUTPUT.
C
C  INPUT : (R*8)  TETA()  = INPUT DATA FILE: ELECTRON TEMPERATURES (EV)
C                           FOR THE DATA-BLOCK BEING ASSESSED.
C                           DIMENSION: ELECTRON TEMPERATURE INDEX
C  INPUT : (R*8)  TEDA()  = INPUT DATA FILE: ELECTRON DENSITIES (CM-3)
C                           FOR THE DATA-BLOCK BEING ASSESSED.
C                           DIMENSION: ELECTRON DENSITY INDEX
C  INPUT : (R*8)  TEVA()  = USER ENTERED: ELECTRON TEMPERATURES (EV)
C                           DIMENSION: TEMPERATURE/DENSITY PAIR INDEX
C  INPUT : (R*8)  DIN()   = USER ENTERED: ELECTRON DENSITIES (CM-3)
C                           DIMENSION: TEMPERATURE/DENSITY PAIR INDEX
C
C
C  INPUT : (R*8)  SXB(,)   =INPUT DATA FILE: FULL SET OF IONIZATIONS PER
C                           PHOTON VALUES FOR THE DATA-BLOCK BEING
C                           ANALYSED.
C                           1ST DIMENSION: ELECTRON TEMPERATURE INDEX
C                           2ND DIMENSION: ELECTRON DENSITY     INDEX
C  OUTPUT: (R*8)  SXBA()  = SPLINE INTERPOLATED OR  EXTRAPOLATED  IONIZ-
C                           ATIONS/PHOTON FOR THE USER ENTERED ELECTRON
C                           TEMPERATURE/DENSITY PAIRS.
C                           DIMENSION: TEMPERATURE/DENSITY PAIR INDEX
C
```

```
C  OUTPUT: (L*4)  LTRNG()=  .TRUE.  => OUTPUT 'SXBA()' VALUE WAS INTER-
C                                     POLATED  FOR  THE  USER  ENTERED
C                                     ELECTRON TEMPERATURE 'TEVA()'.
C                          .FALSE. => OUTPUT 'SXBA()' VALUE WAS EXTRA-
C                                     POLATED  FOR  THE  USER  ENTERED
C                                     ELECTRON TEMPERATURE 'TEVA()'.
C                          DIMENSION: TEMPERATURE/DENSITY PAIR INDEX
C
C  OUTPUT: (L*4)  LDRNG()=  .TRUE.  => OUTPUT 'SXBA()' VALUE WAS INTER-
C                                     POLATED  FOR  THE  USER  ENTERED
C                                     ELECTRON DENSITY 'DIN()'.
C                          .FALSE. => OUTPUT 'SXBA()' VALUE WAS EXTRA-
C                                     POLATED  FOR  THE  USER  ENTERED
C                                     ELECTRON DENSITY 'DIN()'.
C                          DIMENSION: TEMPERATURE/DENSITY PAIR INDEX
C
C          (I*4)  NIN     = PARAMETER = MAX. NO. OF INPUT  TEMP/DENSITY
C                                       VALUES. MUST BE >= 'ITA'&'IDA'
C          (I*4)  NOUT    = PARAMETER = MAX. NO. OF OUTPUT TEMP/DENSITY
C                                       PAIRS.  MUST BE >= 'ITVAL'
C          (I*4)  L1      = PARAMETER = 1
C
C          (I*4)  IED     = ARRAY SUBSCRIPT USED INPUT  FILE  ELECTRON
C                           DENSITIES.
C          (I*4)  IET     = ARRAY SUBSCRIPT USED INPUT  FILE  ELECTRON
C                           TEMPERATURES.
C          (I*4)  IT      = ARRAY  SUBSCRIPT  USED  FOR  USER  ENTERED
C                           TEMPERATURE/DENSITY PAIRS .
C          (I*4)  IOPT    = DEFINES THE BOUNDARY DERIVATIVES FOR THE
C                           SPLINE ROUTINE 'XXSPLE', SEE 'XXSPLE'.
C                           (VALID VALUES = <0, 0, 1, 2, 3, 4)
C
C          (L*4)  LSETX   = .TRUE.  => SET UP SPLINE PARAMETERS RELATING
C                                      TO 'XIN' AXIS.
C                           .FALSE. => DO NOT SET UP SPLINE PARAMETERS
C                                      RELATING TO 'XIN' AXIS.
C                                      (I.E. THEY WERE SET IN A PREVIOUS
C                                            CALL )
C                           (VALUE SET TO .FALSE. BY 'XXSPLE')
C
C          (R*8)  R8FUN1  = FUNCTION - (SEE ROUTINES SECTION BELOW)
C
C          (R*8)  XIN()   = 1) LOG( DATA FILE ELECTRON DENSITIES    )
C                           2) LOG( DATA FILE ELECTRON TEMPERATURES )
C          (R*8)  YIN()   = LOG( DATA FILE IONIZATIONS/PHOTON )
C          (R*8)  XOUT()  = 1) LOG( SCALED USER ENTERED ELECTRON DENS. )
C                           2) LOG( SCALED USER ENTERED ELECTRON TEMPS.)
C          (R*8)  YOUT()  = LOG( OUTPUT GENERATED IONIZATIONS/PHOTON )
C          (R*8)  YPASS(,)= LOG( IONIZATIONS/PHOTON) INTERMEDIATE ARRAY
C                           WHICH   STORES   INTERPOLATED/EXTRAPOLATED
C                           VALUES  BETWEEN  THE  TWO  SPLINE SECTIONS.
C                           SECTIONS.
C          (R*8)  DF()    = SPLINE INTERPOLATED DERIVATIVES
C
C
C NOTE:
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          XXSPLE     ADAS      SPLINE SUBROUTINE (EXTENDED DIAGNOSTICS)
```

```
C          R8FUN1     ADAS      REAL*8 FUNCTION: ( X -> X )
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/37
C          JET EXT. 2520
C
C DATE:    30/04/91
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 02-03-95
C MODIFIED: LALIT JALOTA (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST VERSION
C
C VERSION: 1.2                          DATE: 03-03-95
C MODIFIED: LALIT JALOTA
C               -
C
C VERSION: 1.3                          DATE: 15-04-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - INCREASED PARAMETER NIN 20 -> 26 (CURRENT JET
C                 VERSION HAS 35 BUT THIS WOULD CAUSE PROBLEMS WITH
C                 IDL AT MOMENT).
C
C VERSION: 1.4                          DATE: 22-04-96
C MODIFIED: TIM HAMMOND
C               - INCREASED PARAMETER NIN 26 -> 35 IN LINE WITH JET.
C
C VERSION: 1.5                          DATE: 9-10-97
C MODIFIED: Martin O'Mullane
C               - INCREASED PARAMETER NOUT 20 -> 35.
C
C VERSION: 1.6                          DATE: 9-09-99
C MODIFIED: RICHARD MARTIN
C     - INCREASED NOUT 35 -> 100; NIN 30 -> 100
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IDA,         ITA,         ITVAL,       NDDIM
      INTEGER             NTDIM
      LOGICAL             LDRNG(ITVAL),             LTRNG(ITVAL)
      REAL*8              DIN(ITVAL),  SXB(NTDIM,NDDIM)
      REAL*8              SXBA(ITVAL), TEDA(IDA),   TETA(ITA)
      REAL*8              TEVA(ITVAL)
```

6.5 e1titl: Subroutine e1titl from library adas5xx

```
      SUBROUTINE E1TITL( IBSEL  , DSNAME ,
     &                   ESYM   , IZ     ,
     &                   CWAVEL , CINDM  ,
     &                   TITLX
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: E1TITL *********************
C
C  PURPOSE:  TO CREATE THE DESCRIPTIVE TITLE FOR SELECTED DATA-BLOCK.
C
C  CALLING PROGRAM: ADAS501/SSXB
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  IBSEL    = SELECTED DATA-BLOCK: INDEX
C  INPUT : (C*(*)) DSNAME   = FULL INPUT DATA SET NAME
C
C  INPUT : (C*2)  ESYM     = INPUT DATA SET: EMITTING ION ELEMENT SYMBOL
C  INPUT : (I*4)  IZ       = INPUT DATA SET: EMITTING ION CHARGE-STATE
C
C  INPUT : (C*10) CWAVEL   = SELECTED DATA-BLOCK: WAVELENGTH (ANGS.)
C  INPUT : (C*2)  CINDM    = SELECTED DATA-BLOCK: METASTABLE INDEX
C
C  OUTPUT: (C*120) TITLX   = SELECTED DATA-BLOCK: DESCRIPTIVE TITLE
C
C          (C*2)  C2       = GENERAL USE 2 BYTE  CHARACTER  STRING
C    (I*4)  POS_NOW  = CURRENT POSITION IN TITLE STRING
C    (I*4)  LEN_NAME = LENGTH OF FILENAME
C    (I*4)  IFIRST   = POSITION OF FIRST CHARACTER IN FILENAME
C    (I*4)  ILAST    = POSITION OF LAST CHARACTER IN FILENAME
C
C ROUTINES:
C    XXFCHR = FINDS THE FIRST & LAST OCCURRENCE OF ONE STRING IN
C ANOTHER.
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/37
C          JET EXT. 2620
C
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 02-03-95
C MODIFIED: L. JALOTA (TESSELLA SUPPORT SERVICES PLC)
C               - MODIFY FOR RUNNING UNDER UNIX ENVIRONMENT WITH IDL
C
C VERSION: 1.2                          DATE: 02-03-95
C MODIFIED: L. JALOTA (TESSELLA SUPPORT SERVICES PLC)
C               -
C
C VERSION: 1.3                          DATE: 03-03-95
C MODIFIED: L. JALOTA (TESSELLA SUPPORT SERVICES PLC)
C               -
C
C VERSION: 1.4                          DATE: 06-03-95
C MODIFIED: L. JALOTA (TESSELLA SUPPORT SERVICES PLC)
C               -
C
```

```
C VERSION: 1.5                          DATE: 13-03-95
C MODIFIED: L. JALOTA (TESSELLA SUPPORT SERVICES PLC)
C               -
C
C VERSION: 1.6                          DATE: 13-03-95
C MODIFIED: L. JALOTA (TESSELLA SUPPORT SERVICES PLC)
C               -
C
C VERSION: 1.7                          DATE: 25-05-95
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - COMMENTED OUT LINE DEFINING XXSLEN AND XXFCHR
C                 AS INTEGERS.
C
C VERSION: 1.8                          DATE: 06-02-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - TIDIED UP COMMENTS AND REMOVED SUPERFLUOUS VARIABLES
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      CHARACTER*2         CINDM
      CHARACTER*10        CWAVEL
      CHARACTER*(*)       DSNAME
      CHARACTER*2         ESYM
      CHARACTER*120       TITLX
      INTEGER             IBSEL,       IZ
```

6.6 e2chkb: Subroutine e2chkb from library adas5xx

```
      SUBROUTINE E2CHKB( IUNIT  , NBSEL  , IBSEL  ,
     &                   IZ0IN  , IZ0    ,
     &                   LOPEN  , IRCODE
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: E2CHKB *********************
C
C  PURPOSE: TO CHECK THE SELECTED BLOCK (IBSEL) OF DATA  EXISTS  IN  THE
C           INPUT DATA SET AND IF SO IT REPRESENTS THE ENTERED VALUES OF
C           'IZ0IN' (NUCLEAR CHARGE OF SELECTED IONISING ION ELEMENT).
C
C           IT ALSO CLOSES THE INPUT DATA SET ALLOCATION IF OPEN.
C
C  CALLING PROGRAM: SSZD
C
C  SUBROUTINE:
C
C  INPUT : (I*4)   IUNIT   = UNIT TO WHICH INPUT DATA SET IS ALLOCATED
C  INPUT : (I*4)   NBSEL   = TOTAL NUMBER OF DATA-BLOCKS READ FROM INPUT
C                            DATA SET.
C  INPUT : (I*4)   IBSEL   = INDEX OF DATA-BLOCK SELECTED FOR ANALYSIS
C
C  INPUT : (I*4)   IZ0IN   = REQUESTED: NUCLEAR CHARGE OF ELEMENT
C  INPUT : (I*4)   IZ0     = INPUT FILE: NUCLEAR CHARGE OF ELEMENT
C
C  I/O   : (L*4)   LOPEN   = INPUT : .TRUE.  => INPUT DATA SET OPEN.
C                                    .FALSE. => INPUT DATA SET CLOSED.
C                            OUTPUT: ALWAYS RETURNED AS .FALSE.
C  OUTPUT: (I*4)   IRCODE  = RETURN CODE FROM SUBROUTINE:
C                            0 => NO ERROR DETECTED.
C                            2 => DISCREPANCY BETWEEN REQUESTED ELEMENT
C                                 AND THAT IN INPUT DATA FILE.
C                            3 => SELECTED DATA-BLOCK  OUT OF RANGE  OR
C                                 DOES NOT EXIST.
C
C           (I*4)   I4UNIT  = FUNCTION (SEE ROUTINE SECTION BELOW)
C
CA          (c*80)  DSNAME  = DATA FILE NAME UNDER UNIX, INCLUDING PATH.
C
C ROUTINES:
C           ROUTINE    SOURCE    BRIEF DESCRIPTION
C           ------------------------------------------------------------
C           E2FILE     ADAS      OPEN DATA SET FOR SELECTED ELEMENT
C           I4UNIT     ADAS      FETCH UNIT NUMBER FOR OUTPUT OF MESSAGES
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/37
C          JET EXT. 2620
C
C DATE:    06/06/91
C
C UPDATE:  23/04/93 - PE BRIDEN - ADAS91: ADDED I4UNIT FUNCTION TO WRITE
C                                         STATEMENTS FOR SCREEN MESSAGES
C
C UPDATE:  24/06/93 - PE BRIDEN - ADAS91: CHANGED I4UNIT(0)-> I4UNIT(-1)
C
C UPDATE:  10/11/94 - L. JALOTA - UPDATED TO RUN UNDER UNIX.
C-----------------------------------------------------------------------
```

```
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IBSEL,       IRCODE,      IUNIT,       IZ0
      INTEGER             IZ0IN,       NBSEL
      LOGICAL             LOPEN
```

6.7 e2file: Subroutine e2file from library adas5xx

```
      SUBROUTINE E2FILE( IUNIT , IZ0 , IRCODE , DSNAME )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: E2FILE *********************
C
C  PURPOSE: TO OPEN AN IONIZATION RATE-COEFFT 'IONELEC' DATA SET
C           BY DEFAULT, OR AN ALTERNATIVE DATA SET IF REQUIRED, FOR
C           IONIZING ION WITH NUCLEAR CHARGE 'IZ0'
C           THIS WILL BE CONNECTED TO UNIT 'IUNIT'.
C
C   DATA SET OPENED: $ADASUSER/<DEFADF>/<GROUP>(OPTIONAL)/<TYPE>/
C  <GROUP_EXT>#<ELEMENT SYMBOL>'
C
C  CALLING PROGRAM: SSZD
C
C  SUBROUTINE:
C
C  INPUT : (I*4)   IUNIT   = UNIT TO WHICH DATA SET WILL BE CONNECTED
C  INPUT : (I*4)   IZ0     = NUCLEAR CHARGE OF EMITTING ION REQUESTED
C
C  OUTPUT: (I*4)   IRCODE  = RETURN CODE FROM SUBROUTINE:
C                            0 => DATA SET SUCCESSFULLY CONNECTED
C                            1 => REQUESTED DATA  SET  MEMBER  DOES  NOT
C                                 EXISTS - DATA SET NOT CONNECTED.
C                            9 => REQUESTED DATA  SET  EXISTS BUT CANNOT
C                                 BE OPENED.
C  OUTPUT: (C*80)  DSNAME  = NAME OF OPENED DATA SET UNDER UNIX
C
C          (I*4)   IDLEN   = LENGTH, IN BYTES, OF FIXED 'DSNAME' PREFIX
C          (I*4)   LENF1   = FIRST NON-BLANK CHR OF 'DSNAME' GROUP PART
C          (I*4)   LENF2   = LAST  NON-BLANK CHR OF 'DSNAME' GROUP PART
C          (I*4)   LENF3   = FIRST NON-BLANK CHR OF 'DSNAME' TYPE PART
C          (I*4)   LENF4   = LAST  NON-BLANK CHR OF 'DSNAME' TYPE PART
C          (I*4)   LENF5   = LAST  NON-BLANK CHR OF 'DSNAME' USERID PART
C          (I*4)   LENF6   = LAST  NON-BLANK CHR OF 'DSNAME' USERID PART
C          (I*4)   LENF7   = LAST  NON-BLANK CHR OF 'DSNAME' EXTENSION PART
C          (I*4)   LENF8   = LAST  NON-BLANK CHR OF 'DSNAME' EXTENSION PART
CA         (C*1)   HASH    = '#'   IF NON-BLANK EXT, ELSE ' '.
C          (C*2)   XFESYM  = FUNCTION - (SEE ROUTINES SECTION BELOW)
C          (C*2)   ESYM    = ELEMENT SYMBOL FOR NUCLEAR CHARGE 'IZ0'
C          (C*3)   USREXT  = ADAS SOURCE DATA FILE EXTENSION
CA         (C*80)  USERID  = ADAS SOURCE DATA USER ID
C          (C*8)   USRGRP  = ADAS SOURCE DATA GROUPNAME
CA         (C*80)  USRTYP  = ADAS SOURCE DATA TYPENAME
C    (C*5)   DEFADF  = DEFAULT DATA DIRECTORY, I.E. ADF13
C
C          (L*4)   LEXIST  = .TRUE.  => REQUESTED  DATA  SET  EXISTS.
C                            .FALSE. => REQUESTED  DATA  SET  DOES  NOT
C                                       EXIST.
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C    XXUID      ADAS      FETCHES/SETS ADAS SOURCE DATA USER ID
C          XXSSZD     ADAS      FETCHES/SETS ADAS SOURCE DATA FILENAME
C                               AND FILE EXTENSION
C          XFESYM     ADAS      CHARACTER*2 FUNCTION -
C                               GATHERS ELEMENT SYMBOL FOR NUC. CHARGE
C    XXSLEN     ADAS      FINDS FIRST AND LAST NON-BLANK
```

```
C CHARACTERS IN STRING.
C
C AUTHOR:  H.P. SUMMERS
C    K1/1/57
C    JET EXT. 4941
C DATE:    11/10/91
C
C UPDATE:  10/03/93 - PE BRIDEN: ADDED CALL TO XXUID AND USERID VARIABLE
C                                - NOW ALLOWS ANY INPUT DATASET USER ID.
C UPDATE:   2/09/93 - HPS      : ADDED CALL TO XXSSZD AND USRGRP, USRTYP
C                                  AND USREXT NAMES
C                                - NOW ALLOWS ANY INPUT DATASET FILENAME
C                                  AND EXTENSION
C UPDATE:  23/11/93 - PEB      : CORRECT ERROR - A '.' HAD MISTAKENLY
C                                BEEN PLACED BEFORE THE MEMBER NAME IN
C                                VARIABLE DSNAME.
C
C UPDATE:  10/11/94 - L. JALOTA: MODIFIED CODE FOR RUNNING UNDER UNIX
C  USING NEW FILE NAMING CONVENTION.
C  "ACTION" KEYWORD IN OPEN COMMAND IS IBM
C  SO REMOVED HERE.
C   ADDED DEFADF
C UPDATE:  22/11/94 - L. JALOTA: TIDIED UP CHARACTER LENGTH DEFINITIONS
C
C UPDATE:  24/03/95 - HPS      : INTRODUCED HASH TO ELIMINATE # IN FILE IF
C                                THERE IS NO EXTENSION PART OF THE FILE NAME
C                                ALTER LOGIC TO ALLOW USRTYP, USREXT TO BE A
C                                SINGLE CHARACTER.
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      CHARACTER*80        DSNAME
      INTEGER             IRCODE,      IUNIT,       IZ0
```

6.8 e2spln: Subroutine e2spln from library adas5xx

```
      SUBROUTINE E2SPLN( ITA    , ITVAL   ,
     &                   BWNO   ,
     &                   TETA   , TEVA    ,
     &                   SZD    , SZDA    ,
     &                   LTRNG
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: E2SPLN *********************
C
C  PURPOSE:
C          PERFORMS CUBIC SPLINE ON LOG(TEMPERATURE <EV> ) VERSUS
C          LOG(SCALED IONIZATION RATE COEFFICIENTS).
C          INPUT DATA FOR A GIVEN IONZING ION COMBINATION DATA-BLOCK.
C
C          USING  ONE-WAY SPLINES  IT  CALCULATES  THE  IONIZATION  RATE
C          COEFFICIENT  FOR  'ITVAL'  ELECTRON  TEMPERATURE VALUES  FROM
C          THE LIST OF ELECTRON TEMPERATURES READ IN FROM THE INPUT FILE
C
C          IF A  VALUE  CANNOT  BE  INTERPOLATED  USING  SPLINES  IT  IS
C          EXTRAPOLATED VIA 'XXSPLE'. (SEE NOTES BELOW).
C
C  CALLING PROGRAM: ADAS602/SSZD
C
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  ITA     = INPUT DATA FILE: NUMBER OF ELECTRON TEMPERA-
C                           TURES READ FOR THE DATA-BLOCK BEING ASSESSED
C  INPUT : (I*4)  ITVAL   = NUMBER OF ISPF ENTERED ELECTRON TEMPERATURE
C                           VALUES FOR  WHICH  IOINIZATION RATE COEFFTS
C                           ARE REQUIRED FOR TABULAR/GRAPHICAL OUTPUT.
C
C  INPUT : (R*8)  BWNO    = INPUT DATA FILE: IONIZATION POTENTIAL (cm-1)
C                           FOR THE DATA-BLOCK BEING ASSESSED.
C
C  INPUT : (R*8)  TETA()  = INPUT DATA FILE: ELECTRON TEMPERATURES (EV)
C                           FOR THE DATA-BLOCK BEING ASSESSED.
C                           DIMENSION: ELECTRON TEMPERATURE INDEX
C  INPUT : (R*8)  TEVA()  = USER ENTERED: ELECTRON TEMPERATURES (EV)
C                           DIMENSION: TEMPERATURE/DENSITY PAIR INDEX
C
C  INPUT : (R*8)  SZD()    =INPUT DATA FILE: FULL SET OF ZERO DENSITY
C                           IONIZATION RATE COEFFTS FOR THE DATA-BLOCK
C                           BEING ANALYSED.
C                           1ST DIMENSION: ELECTRON TEMPERATURE INDEX
C  OUTPUT: (R*8)  SZDA()  = SPLINE INTERPOLATED OR  EXTRAPOLATED  ZERO
C                           DENSITY IONIZATION RATE COEFFICIENTS FOR
C                           THE USER ENTERED ELECTRON TEMPERATURES.
C                           DIMENSION: ELECTRON TEMPERATURE INDEX
C
C  OUTPUT: (L*4)  LTRNG()=  .TRUE.  => OUTPUT 'SZDA()' VALUE WAS INTER-
C                                      POLATED  FOR  THE  USER  ENTERED
C                                      ELECTRON TEMPERATURE 'TEVA()'.
C                           .FALSE. => OUTPUT 'SZDA()' VALUE WAS EXTRA-
C                                      POLATED  FOR  THE  USER  ENTERED
C                                      ELECTRON TEMPERATURE 'TEVA()'.
C                           DIMENSION: ELECTRON TEMPERATURE INDEX
C
```

```
C          (I*4)  NIN     = PARAMETER = MAX. NO. OF INPUT  TEMPERATURE
C                                       VALUES. MUST BE >= 'ITA'
C          (I*4)  NOUT    = PARAMETER = MAX. NO. OF OUTPUT TEMPERATURE
C                                       PAIRS.  MUST BE >= 'ITVAL'
C          (I*4)  L1      = PARAMETER = 1
C
C          (R*8)  BCONST  = PARAMETER = 1/(SCALED BOLTZMANN'S CONSTANT)
C
C          (I*4)  IET     = ARRAY SUBSCRIPT USED INPUT  FILE  ELECTRON
C                           TEMPERATURES.
C          (I*4)  IT      = ARRAY  SUBSCRIPT  USED  FOR  USER  ENTERED
C                           TEMPERATURE VALUES.
C          (I*4)  IOPT    = DEFINES THE BOUNDARY DERIVATIVES FOR THE
C                           SPLINE ROUTINE 'XXSPLE', SEE 'XXSPLE'.
C                           (VALID VALUES = <0, 0, 1, 2, 3, 4)
C
C          (L*4)  LSETX   = .TRUE.  => SET UP SPLINE PARAMETERS RELATING
C                                      TO 'XIN' AXIS.
C                           .FALSE. => DO NOT SET UP SPLINE PARAMETERS
C                                      RELATING TO 'XIN' AXIS.
C                                      (I.E. THEY WERE SET IN A PREVIOUS
C                                            CALL )
C                           (VALUE SET TO .FALSE. BY 'XXSPLE')
C
C          (R*8)  R8FUN1  = FUNCTION - (SEE ROUTINES SECTION BELOW)
C          (R*8)  SCONST  = SCALING CONSTANT USED TO SCALE THE IONIZA-
C                           TION RATE COEFFT. WHEN SPLINNING.
C                           = IONIZATION POTENTIAL / BOLTZMANN CONST.
C
C          (R*8)  XIN()   = LOG( DATA FILE ELECTRON TEMPERATURES )
C          (R*8)  YIN()   = LOG( DATA FILE SCALED ION. RATE COEFFTS.)
C          (R*8)  XOUT()  = LOG( USER ENTERED ELECTRON TEMPS.)
C          (R*8)  YOUT()  = LOG( OUTPUT GENERATED SCALED ION. RATE COEF)
C          (R*8)  DF()    = SPLINE INTERPOLATED DERIVATIVES
C
C
C NOTE:
C
C          ONE-DIMENSIONAL SPLINE CARRIED OUT BY THIS SUBROUTINE:
C
C           LOG( EXP(<ion.pt.>/<k>.<Te>) . Szd )  vs. LOG( Te )
C
C           ion.pt. = ionization potential (units: cm-1)
C           k       = Boltzmann's constant (= 1/1.23977E-04)
C           Te      = electron temperature (units: eV)
C           Szd     = zero density ionization rate coefficient
C                                      (units: cm**3/sec)
C
C           Extrapolation criteria:
C
C           Low  Te: zero gradient  extrapolation (i.e. DY(1)  = 0.0)
C           High Te: zero curvature extrapolation (i.e. DDY(N) = 0.0)
C
C           (These criteria are met by calling XXSPLE with IOPT=4)
C
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          XXSPLE     ADAS      SPLINE SUBROUTINE (EXTENDED DIAGNOSTICS)
C          R8FUN1     ADAS      REAL*8 FUNCTION: ( X -> X )
```

```
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/37
C          JET EXT. 2620
C
C DATE:    07/06/91
C
C VERSION: 1.1 DATE: ???
C MODIFIED: ???
C     - FIRST PUT UNDER SCCS
C
C VERSION: 1.2 DATE: 23/05/96
C MODIFIED: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C     - INCREASED NOUT TO 35
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             ITA,         ITVAL
      LOGICAL             LTRNG(ITVAL)
      REAL*8              BWNO,        SZD(ITA),    SZDA(ITVAL)
      REAL*8              TETA(ITA),   TEVA(ITVAL)
```

6.9 e2titl: Subroutine e2titl from library adas5xx

```
      SUBROUTINE E2TITL( IBSEL  , DSNAME ,
     &                   CICODE , CFCODE ,
     &                   CIION  , CFION  ,
     &                   TITLX
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: E2TITL *********************
C
C  PURPOSE:  TO CREATE THE DESCRIPTIVE TITLE FOR SELECTED DATA-BLOCK.
C
C  CALLING PROGRAM: ADAS502/SSZD
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  IBSEL    = SELECTED DATA-BLOCK: INDEX
C  INPUT : (C*80) DSNAME   = DATA FILE NAME UNDER UNIX INCLUDING PATH
C
C  INPUT : (C*2)  CICODE   = SELECTED DATA-BLOCK: INITIAL STATE
C                                                 METASTABLE INDEX.
C  INPUT : (C*2)  CFCODE   = SELECTED DATA-BLOCK: FINAL   STATE
C                                                 METASTABLE INDEX.
C
C  INPUT : (C*5)  CIION    = SELECTED DATA-BLOCK: INITIAL ION
C  INPUT : (C*5)  CFION    = SELECTED DATA-BLOCK: FINAL   ION
C
C  OUTPUT: (C*80) TITLX    = SELECTED DATA-BLOCK: DESCRIPTIVE TITLE
C
C  GENERAL :
C          (C*2)  C2       = GENERAL USE 2 BYTE  CHARACTER  STRING
C
C    (I*4)  START    = POSITION OF FIRST NON_BLANK CHARACTER IN
C       DSNAME
C    (I*4)  END      = POSITION OF LAST  NON_BLANK CHARACTER IN
C       DSNAME
C
C ROUTINES:
CA    XXSLEN          = UTILITY ROUTINE WHICH FINDS FIRST AND LAST
CA      NON-BLANK CHARACTERS IN A STRING.
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/37
C          JET EXT. 2620
C
C DATE:    07/06/91
C
C UPDATE : L. JALOTA  10/11/94 - MODIFIED TO RUN UNDER UNIX.
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      CHARACTER*2         CFCODE
      CHARACTER*5         CFION
      CHARACTER*2         CICODE
      CHARACTER*5         CIION
      CHARACTER*(*)       DSNAME
      CHARACTER*120       TITLX
      INTEGER             IBSEL
```

6.10 e3chkb: Subroutine e3chkb from library adas5xx

```
      SUBROUTINE E3CHKB( IUNIT  , NBSEL  , IBSEL  ,
     &                   IZ0IN  , IZIN   ,
     &                   IZ0    , IZ     ,
     &                   LOPEN  , IRCODE
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: E3CHKB *********************
C
C  PURPOSE: TO CHECK THE SELECTED BLOCK (IBSEL) OF DATA  EXISTS  IN  THE
C           INPUT DATA SET.
C
C           IF SO IT REPRESENTS THE ENTERED VALUES OF
C           'IZ0IN' (NUCLEAR CHARGE OF EMITTING ION)          &
C           'IZIN'  (CHARGE OF EMITTING ION)
C
C           IT ALSO CLOSES THE INPUT DATA SET ALLOCATION IF OPEN.
C
C  CALLING PROGRAM: SPEC
C
C  SUBROUTINE:
C
C  INPUT : (I*4)   IUNIT   = UNIT TO WHICH INPUT DATA SET IS ALLOCATED
C  INPUT : (I*4)   NBSEL   = TOTAL NUMBER OF DATA-BLOCKS READ FROM INPUT
C                            DATA SET.
C  INPUT : (I*4)   IBSEL   = INDEX OF DATA-BLOCK SELECTED FOR ANALYSIS
C
C  INPUT : (I*4)   IZ0IN   = REQUESTED: NUCLEAR CHARGE OF EMITTING ION
C  INPUT : (I*4)   IZIN    = REQUESTED: CHARGE OF EMITTING ION
C
C  INPUT : (I*4)   IZ0     = INPUT FILE: NUCLEAR CHARGE OF EMITTING ION
C  INPUT : (I*4)   IZ      = INPUT FILE: CHARGE OF EMITTING ION
C
C  I/O   : (L*4)   LOPEN   = INPUT : .TRUE.  => INPUT DATA SET OPEN.
C                                    .FALSE. => INPUT DATA SET CLOSED.
C                            OUTPUT: ALWAYS RETURNED AS .FALSE.
C  OUTPUT: (I*4)   IRCODE  = RETURN CODE FROM SUBROUTINE:
C                            0 => NO ERROR DETECTED.
C                            2 => DISCREPANCY BETWEEN REQUESTED CHARGES
C                                 AND THOSE IN INPUT DATA FILE.
C                            3 => SELECTED DATA-BLOCK  OUT OF RANGE  OR
C                                 DOES NOT EXIST.
C
C           (I*4)   I4UNIT  = FUNCTION (SEE ROUTINE SECTION BELOW)
C
C           (C*80)  DSNAME  = UNIX NAME OF DATA SET OPENED
C
C ROUTINES:
C           ROUTINE    SOURCE    BRIEF DESCRIPTION
C           ------------------------------------------------------------
C           E3FILE     ADAS      OPEN DATA SET FOR SELECTED EMITTER
C           I4UNIT     ADAS      FETCH UNIT NUMBER FOR OUTPUT OF MESSAGES
C
C AUTHOR:  H. P. SUMMERS
C          K1/1/57
C          JET EXT. 4941
C
C DATE:    11/10/91
C
```

```
C UPDATE:  23/04/93 - PE BRIDEN - ADAS91: ADDED I4UNIT FUNCTION TO WRITE
C                                         STATEMENTS FOR SCREEN MESSAGES
C
C UPDATE:  24/05/93 - PE BRIDEN - ADAS91: CHANGED I4UNIT(0)-> I4UNIT(-1)
C
C UPDATE:   1/11/94 - L. JALOTA - UPDATED TO RUN UNDER UNIX ON DEC-ALPHA
C
C-----------------------------------------------------------------------
C
C VERSION: 1.1 DATE: 25-05-95
C MODIFIED: UNKNOWN (SOMEONE FROM TESSELLA SUPPORT SERVICES PLC)
C     - PUT UNDER SCCS CONTROL
C
C VERSION: 1.2 DATE: 20-07-07
C MODIFIED: Allan Whiteford
C     - Small modification to comments to allow for automatic
C             documentation preparation.
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IBSEL,       IRCODE,      IUNIT,       IZ
      INTEGER             IZ0,         IZ0IN,       IZIN,        NBSEL
      LOGICAL             LOPEN
```

6.11 e3data: Subroutine e3data from library adas5xx

```
       SUBROUTINE E3DATA( IUNIT  , DSNAME ,
     &                    NSTORE , NTDIM  , NDDIM  ,
     &                    IZ0    , IZ     , IZ1    , ESYM  ,
     &                    NBSEL  , ISELA  ,
     &                    CWAVEL , CFILE  , CTYPE  , CINDM ,
     &                    ITA    , IDA    ,
     &                    TETA   , TEDA   ,
     &                    PEC
     &                  )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: E3DATA *********************
C
C  PURPOSE:  TO  FETCH  DATA  FROM  INPUT PHOTON EMISSIVITY FILE
C            FOR A GIVEN EMITTING ION (ELEMENT AND CHARGE).
C            (MEMBER STORED IN IONELEC.DATA - MEMBER PREFIX 'PEC#').
C
C  CALLING PROGRAM: ADAS503/SPEC
C
C  DATA:
C
C           UP TO 'NSTORE' SETS (DATA-BLOCKS) OF DATA MAY BE  READ FROM
C           THE FILE - EACH BLOCK FORMING A COMPLETE SET OF IONIZATIONS
C           PER PHOTON VALUES FOR GIVEN TEMP./DENSITY COMBINATION. EACH
C           DATA-BLOCK  IS  ANALYSED INDEPENDENTLY OF ANY  OTHER  DATA-
C           BLOCK.
C
C           THE UNITS USED IN THE DATA FILE ARE TAKEN AS FOLLOWS:
C
C           TEMPERATURES        : EV
C           DENSITIES           : CM-3
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  IUNIT    = UNIT TO WHICH INPUT FILE IS ALLOCATED.
CA INPUT : (C*80) DSNAME   = NAME OF DATA FILE BEING READ
C
C  INPUT : (I*4)  NSTORE   = MAXIMUM NUMBER  OF  INPUT DATA-BLOCKS  THAT
C                            CAN BE STORED.
C  INPUT : (I*4)  NTDIM    = MAX NUMBER OF ELECTRON TEMPERATURES ALLOWED
C  INPUT : (I*4)  NDDIM    = MAX NUMBER OF ELECTRON DENSITIES    ALLOWED
C
C  OUTPUT: (I*4)  IZ0      = READ - EMITTING ION - NUCLEAR CHARGE
C  OUTPUT: (I*4)  IZ       = READ - EMITTING ION - CHARGE
C  OUTPUT: (I*4)  IZ1      = READ - EMITTING ION - CHARGE + 1
C  OUTPUT: (C*2)  ESYM     = READ - EMITTING ION - ELEMENT SYMBOL
C
C  OUTPUT: (I*4)  NBSEL    = NUMBER OF DATA-BLOCKS ACCEPTED & READ IN.
C  OUTPUT: (I*4)  ISELA()  = READ - DATA-SET DATA-BLOCK ENTRY INDICES
C                            DIMENSION: DATA-BLOCK INDEX
C
C  OUTPUT: (C*10) CWAVEL() = READ - WAVELENGTH (ANGSTROMS)
C                            DIMENSION: DATA-BLOCK INDEX
C  OUTPUT: (C*8)  CFILE()  = READ - SPECIFIC ION FILE SOURCE
C                            DIMENSION: DATA-BLOCK INDEX
C  OUTPUT: (C*8)  CTYPE()  = READ - DATA TYPE
C                            DIMENSION: DATA-BLOCK INDEX
C  OUTPUT: (C*2)  CINDM()  = READ - METASTABLE INDEX
C                            DIMENSION: DATA-BLOCK INDEX
```

```
C
C  OUTPUT: (I*4)  ITA()    = READ - NUMBER OF ELECTRON TEMPERATURES
C                            DIMENSION: DATA-BLOCK INDEX
C  OUTPUT: (I*4)  IDA()    = READ - NUMBER OF ELECTRON DENSITIES
C                            DIMENSION: DATA-BLOCK INDEX
C
C  OUTPUT: (R*8)  TETA(,)  = READ - ELECTRON TEMPERATURES (UNITS: eV)
C                            1st DIMENSION: ELECTRON TEMPERATURE INDEX
C                            2nd DIMENSION: DATA-BLOCK INDEX
C  OUTPUT: (R*8)  TEDA(,)  = READ - ELECTRON DENSITIES (UNITS: CM-3)
C                            1st DIMENSION: ELECTRON DENSITY INDEX
C                            2nd DIMENSION: DATA-BLOCK INDEX
C
C  OUTPUT: (R*8)  PEC(,,)   =READ - PHOTON EMISSIVITY VALUES
C                            1st DIMENSION: ELECTRON TEMPERATURE INDEX
C                            2nd DIMENSION: ELECTRON DENSITY INDEX
C                            3rd DIMENSION: DATA-BLOCK INDEX
C
C          (I*4)  I4EIZ0   = FUNCTION - (SEE ROUTINES SECTION BELOW)
C          (I*4)  I4FCTN   = FUNCTION - (SEE ROUTINES SECTION BELOW)
C          (I*4)  I4UNIT   = FUNCTION - (SEE ROUTINES SECTION BELOW)
C          (I*4)  IBLK     = ARRAY INDEX: DATA-BLOCK INDEX
C          (I*4)  ITT      = ARRAY INDEX: ELECTRON TEMPERATURE INDEX
C          (I*4)  ITD      = ARRAY INDEX: ELECTRON DENSITY     INDEX
C          (I*4)  NTNUM    = NUMBER OF ELECTRON TEMPERATURES FOR CURRENT
C                            DATA-BLOCK
C          (I*4)  NDNUM    = NUMBER OF ELECTRON DENSITIES    FOR CURRENT
C                            DATA-BLOCK
C          (I*4)  IABT     = RETURN CODE FROM 'I4FCTN'
C          (I*4)  IPOS1    = GENERAL USE STRING INDEX VARIABLE
C          (I*4)  IPOS2    = GENERAL USE STRING INDEX VARIABLE
C
C          (L*4)  LBEND    = IDENTIFIES WHETHER THE LAST OF THE  INPUT
C                            DATA SUB-BLOCKS HAS BEEN LOCATED.
C                            (.TRUE. => END OF SUB-BLOCKS REACHED)
C
C          (C*1)  CSLASH   = '/' - DELIMITER FOR 'XXHKEY'
C          (C*2)  C2       = GENERAL USE TWO BYTE CHARACTER STRING
C          (C*5)  IONNAM   = EMITTING ION READ FROM DATASET
C          (C*6)  CKEY1    = 'FILMEM' - INPUT BLOCK HEADER KEY
C          (C*4)  CKEY2    = 'TYPE  ' - INPUT BLOCK HEADER KEY
C          (C*4)  CKEY3    = 'INDM  ' - INPUT BLOCK HEADER KEY
C          (C*4)  CKEY4    = 'ISEL  ' - INPUT BLOCK HEADER KEY
C          (C*80) C80      = GENERAL USE 80 BYTE  CHARACTER  STRING  FOR
C                            THE INPUT OF DATA-SET RECORDS.
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          XXHKEY     ADAS      OBTAIN KEY/RESPONSE STRINGS FROM TEXT
C          I4EIZ0     ADAS      INTEGER*4 FUNCTION    -
C                               RETURNS Z0 FOR GIVEN ELEMENT SYMBOL
C          I4FCTN     ADAS      INTEGER*4 FUNCTION    -
C                               CONVERT CHARACTER STRING TO INTEGER
C          I4UNIT     ADAS      INTEGER*4 FUNCTION    -
C                               FETCH UNIT NUMBER FOR OUTPUT OF MESSAGES
C
C AUTHOR:  H. P. SUMMERS
C          K1/1/57
C          JET EXT. 4941
C
```

```
C DATE:    11/10/91
C
C UPDATE:  05/12/91 - PE BRIDEN: IONNAM NOW ALLOWED TO OCCUPY EITHER
C                                4 OR 5 SPACES IN THE HEADER.
C
C UPDATE:  23/04/93 - PE BRIDEN - ADAS91: ADDED I4UNIT FUNCTION TO WRITE
C                                         STATEMENTS FOR SCREEN MESSAGES
C
C UPDATE:  24/05/93 - PE BRIDEN - ADAS91: CHANGED I4UNIT(0)-> I4UNIT(-1)
C
C UPDATE:  27/2/95  - L. JALOTA - IDL_ADAS : INCREASED SIZE DSNAME FOR
C      USE UNDER UNIX SYSTEMS
C
C UNIX-IDL PORT:
C
C VERSION: 1.2                          DATE: 23-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - CORRECTED FORMAT STATEMENTS FOR DSNAME LENGTH
C
C-----------------------------------------------------------------------
      CHARACTER*8         CFILE(NSTORE)
      CHARACTER*2         CINDM(NSTORE)
      CHARACTER*8         CTYPE(NSTORE)
      CHARACTER*10        CWAVEL(NSTORE)
      CHARACTER*80        DSNAME
      CHARACTER*2         ESYM
      INTEGER             IDA(NSTORE), ISELA(NSTORE)
      INTEGER             ITA(NSTORE), IUNIT,       IZ,          IZ0
      INTEGER             IZ1,         NBSEL,       NDDIM,       NSTORE
      INTEGER             NTDIM
      REAL*8              PEC(NTDIM,NDDIM,NSTORE),  TEDA(NDDIM,NSTORE)
      REAL*8              TETA(NTDIM,NSTORE)
```

6.12 e3file: Subroutine e3file from library adas5xx

```
      SUBROUTINE E3FILE( IUNIT , IZ0 , IZ , IRCODE , DSNAME )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: E3FILE *********************
C
C  PURPOSE: TO OPEN A PHOTON EMISSIVITY 'IONELEC' DATA SET
C           BY DEFAULT, OR AN ALTERNATIVE DATA SET IF REQUIRED, FOR
C           EMITTING ION WITH NUCLEAR CHARGE 'IZ0' AND CHARGE 'IZ'.
C           THIS WILL BE CONNECTED TO UNIT 'IUNIT'.
C
C   DATA SET OPENED: $ADASUSER/<DEFADF>/<GROUP>(OPTIONAL)/<TYPE>/
C  <GROUP_EXT>#<ELEMENT SYMBOL><CHARGE>
C
C  CALLING PROGRAM: SPEC
C
C  SUBROUTINE:
C
C  INPUT : (I*4)   IUNIT   = UNIT TO WHICH DATA SET WILL BE CONNECTED
C  INPUT : (I*4)   IZ0     = NUCLEAR CHARGE OF EMITTING ION REQUESTED
C  INPUT : (I*4)   IZ      = ION CHARGE OF EMITTING ION REQUESTED
C
C  OUTPUT: (I*4)   IRCODE  = RETURN CODE FROM SUBROUTINE:
C                            0 => DATA SET SUCCESSFULLY CONNECTED
C                            1 => REQUESTED DATA  SET  MEMBER  DOES  NOT
C                                 EXISTS - DATA SET NOT CONNECTED.
C                            6 => INVALID VALUE FOR 'IZ' ENTERED.
C                                 ( 0 <= 'IZ' <= 99 )
C                            9 => REQUESTED DATA  SET  EXISTS BUT CANNOT
C                                 BE OPENED.
C  OUTPUT: (C*80)  DSNAME  = NAME OF OPENED DATA SET UNDER UNIX
C
C          (I*4)   IDLEN   = LENGTH, IN BYTES, OF FIXED 'DSNAME' PREFIX
C          (I*4)   LENF1   = FIRST NON-BLANK CHR OF 'DSNAME' GROUP PART
C          (I*4)   LENF2   = LAST  NON-BLANK CHR OF 'DSNAME' GROUP PART
C          (I*4)   LENF3   = FIRST NON-BLANK CHR OF 'DSNAME' TYPE PART
C          (I*4)   LENF4   = LAST  NON-BLANK CHR OF 'DSNAME' TYPE PART
C          (I*4)   LENF5   = LAST  NON-BLANK CHR OF 'DSNAME' USERID PART
C          (I*4)   LENF6   = LAST  NON-BLANK CHR OF 'DSNAME' USERID PART
C          (I*4)   LENF7   = LAST  NON-BLANK CHR OF 'DSNAME' EXTENSION PART
C          (I*4)   LENF8   = LAST  NON-BLANK CHR OF 'DSNAME' EXTENSION PART
C          (I*4)   IZEND   = LAST BYTE WRITTEN TO IN 'CZ'. (= 1 OR 2)
C          (C*1)   HASH    = '#' IF NON-BLANK EXT, ELSE ' '.
C          (C*2)   CZ      = 'IZ' (NO LEADING BLANKS)
C          (C*2)   XFESYM  = FUNCTION - (SEE ROUTINES SECTION BELOW)
C          (C*2)   ESYM    = ELEMENT SYMBOL FOR NUCLEAR CHARGE 'IZ0'
CA         (C*3)   USREXT  = ADAS SOURCE DATA FILE EXTENSION
CA         (C*80)  USERID  = ADAS SOURCE DATA USER ID
CA         (C*8)   USRGRP  = ADAS SOURCE DATA GROUPNAME
CA         (C*80)  USRTYP  = ADAS SOURCE DATA TYPENAME
C    (C*6)   DEFADF  = DEFAULT DATA DIRECTORY, I.E. ADF13
C
C          (L*4)   LEXIST  = .TRUE.  => REQUESTED  DATA  SET  EXISTS.
C                            .FALSE. => REQUESTED  DATA  SET  DOES  NOT
C                                       EXIST.
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C    XXUID      ADAS      FETCHES/SETS ADAS SOURCE DATA USER ID
```

```
C          XXSPEC     ADAS      FETCHES/SETS ADAS SOURCE DATA FILENAME
C                               AND FILE EXTENSION
C          XFESYM     ADAS      CHARACTER*2 FUNCTION -
C                               GATHERS ELEMENT SYMBOL FOR NUC. CHARGE
C    XXSLEN     ADAS      FINDS FIRST AND LAST NON-BLANK
C CHARACTERS IN STRING.
C
C AUTHOR:  H.P. SUMMERS
C    K1/1/67
C    JET EXT. 4941
C DATE:    11/10/91
C
C UPDATE:  10/03/93 - PE BRIDEN: ADDED CALL TO XXUID AND USERID VARIABLE
C                                - NOW ALLOWS ANY INPUT DATASET USER ID.
C UPDATE:   2/09/93 - HPS      : ADDED CALL TO XXSPEC AND USRGRP, USRTYP
C                                  AND USREXT NAMES
C                                - NOW ALLOWS ANY INPUT DATASET FILENAME
C                                  AND EXTENSION
C UPDATE:  23/11/93 - PEB      : CORRECT ERROR - A '.' HAD MISTAKENLY
C                                BEEN PLACED BEFORE THE MEMBER NAME IN
C                                VARIABLE DSNAME.
C
C UPDATE:   1/11/94 - L. JALOTA: MODIFIED CODE FOR RUNNING UNDER UNIX
C  USING NEW FILE NAMING CONVENTION.
C  "ACTION" KEYWORD IN OPEN COMMAND IS IBM
C  SO REMOVED HERE.
C UPDATE:   8/11/94 - L. JALOTA: ADDED DEFADF
C
C UPDATE:  22/11/94 - L. JALOTA: TIDIED UP CHARACTER STRING LENGTHS
C
C UPDATE:  22/03/95 _ HPS      : INTRODUCED HASH TO ELIMINATE # IN FILE IF
C                                THERE IS NO EXTENSION PART OF THE FILE NAME
C                                ALTER LOGIC TO ALLOW USRTYP, USREXT TO BE A
C                                SINGLE CHARACTER.
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      CHARACTER*80        DSNAME
      INTEGER             IRCODE,      IUNIT,       IZ,          IZ0
```

6.13 e3spln: Subroutine e3spln from library adas5xx

```
C
      SUBROUTINE E3SPLN( NTDIM  , NDDIM  ,
     &                   ITA    , IDA    , ITVAL   ,
     &                   TETA   , TEDA   , TEVA    , DIN    ,
     &                   PEC    ,          PECA    ,
     &                                     LTRNG   , LDRNG
     &                 )
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: E3SPLN *********************
C
C  PURPOSE:
C          PERFORMS CUBIC SPLINE ON LOG(TEMPERATURE AND DENSITY)
C          VERSUS LOG(IONIZATIONS PER PHOTON)
C          INPUT DATA FOR A GIVEN WAVELENGTH DATA-BLOCK.
C
C          USING  TWO-WAY SPLINES IT CALCULATES  THE  PHOTON EMISSIVITY
C          FOR  'ITVAL'  PAIRS OF  ELECTRON TEMPERATURES  AND  DENSITIES
C          FROM THE TWO-DIMENSIONAL TABLE OF TEMPERATURES/DENSITIES READ
C          IN FROM THE INPUT FILE. IF A  VALUE  CANNOT  BE  INTERPOLATED
C          USING SPLINES IT IS EXTRAPOLATED VIA 'XXSPLE'.
C
C  CALLING PROGRAM: ADAS503/SPEC
C
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  NTDIM   = MAX NUMBER OF ELECTRON TEMPERATURES ALLOWED
C  INPUT : (I*4)  NDDIM   = MAX NUMBER OF ELECTRON DENSITIES    ALLOWED
C
C  INPUT : (I*4)  ITA     = INPUT DATA FILE: NUMBER OF ELECTRON TEMPERA-
C                           TURES READ FOR THE DATA-BLOCK BEING ASSESSED
C  INPUT : (I*4)  IDA     = INPUT DATA FILE: NUMBER OF ELECTRON DENSIT-
C                           IES   READ FOR THE DATA-BLOCK BEING ASSESSED
C  INPUT : (I*4)  ITVAL   = NUMBER OF ISPF ENTERED TEMPERATURE/DENSITY
C                           PAIRS  FOR  WHICH  IOINIZATIONS PER PHOTON
C                           ARE REQUIRED FOR TABULAR/GRAPHICAL OUTPUT.
C
C  INPUT : (R*8)  TETA()  = INPUT DATA FILE: ELECTRON TEMPERATURES (EV)
C                           FOR THE DATA-BLOCK BEING ASSESSED.
C                           DIMENSION: ELECTRON TEMPERATURE INDEX
C  INPUT : (R*8)  TEDA()  = INPUT DATA FILE: ELECTRON DENSITIES (CM-3)
C                           FOR THE DATA-BLOCK BEING ASSESSED.
C                           DIMENSION: ELECTRON DENSITY INDEX
C  INPUT : (R*8)  TEVA()  = USER ENTERED: ELECTRON TEMPERATURES (EV)
C                           DIMENSION: TEMPERATURE/DENSITY PAIR INDEX
C  INPUT : (R*8)  DIN()   = USER ENTERED: ELECTRON DENSITIES (CM-3)
C                           DIMENSION: TEMPERATURE/DENSITY PAIR INDEX
C
C
C  INPUT : (R*8)  PEC(,)   =INPUT DATA FILE: FULL SET OF IONIZATIONS PER
C                           PHOTON VALUES FOR THE DATA-BLOCK BEING
C                           ANALYSED.
C                           1ST DIMENSION: ELECTRON TEMPERATURE INDEX
C                           2ND DIMENSION: ELECTRON DENSITY     INDEX
C  OUTPUT: (R*8)  PECA()  = SPLINE INTERPOLATED OR  EXTRAPOLATED  IONIZ-
C                           ATIONS/PHOTON FOR THE USER ENTERED ELECTRON
C                           TEMPERATURE/DENSITY PAIRS.
```

```
C                          DIMENSION: TEMPERATURE/DENSITY PAIR INDEX
C
C  OUTPUT: (L*4)  LTRNG()=  .TRUE.  => OUTPUT 'PECA()' VALUE WAS INTER-
C                                      POLATED  FOR  THE  USER  ENTERED
C                                      ELECTRON TEMPERATURE 'TEVA()'.
C                          .FALSE. => OUTPUT 'PECA()' VALUE WAS EXTRA-
C                                      POLATED  FOR  THE  USER  ENTERED
C                                      ELECTRON TEMPERATURE 'TEVA()'.
C                          DIMENSION: TEMPERATURE/DENSITY PAIR INDEX
C
C  OUTPUT: (L*4)  LDRNG()=  .TRUE.  => OUTPUT 'PECA()' VALUE WAS INTER-
C                                      POLATED  FOR  THE  USER  ENTERED
C                                      ELECTRON DENSITY 'DIN()'.
C                          .FALSE. => OUTPUT 'PECA()' VALUE WAS EXTRA-
C                                      POLATED  FOR  THE  USER  ENTERED
C                                      ELECTRON DENSITY 'DIN()'.
C                          DIMENSION: TEMPERATURE/DENSITY PAIR INDEX
C
C          (I*4)  NIN     = PARAMETER = MAX. NO. OF INPUT  TEMP/DENSITY
C                                        VALUES. MUST BE >= 'ITA'&'IDA'
C          (I*4)  NOUT    = PARAMETER = MAX. NO. OF OUTPUT TEMP/DENSITY
C                                        PAIRS.  MUST BE >= 'ITVAL'
C          (I*4)  L1      = PARAMETER = 1
C
C          (I*4)  IED     = ARRAY SUBSCRIPT USED INPUT  FILE  ELECTRON
C                           DENSITIES.
C          (I*4)  IET     = ARRAY SUBSCRIPT USED INPUT  FILE  ELECTRON
C                           TEMPERATURES.
C          (I*4)  IT      = ARRAY  SUBSCRIPT  USED  FOR  USER  ENTERED
C                           TEMPERATURE/DENSITY PAIRS .
C          (I*4)  IOPT    = DEFINES THE BOUNDARY DERIVATIVES FOR THE
C                           SPLINE ROUTINE 'XXSPLE', SEE 'XXSPLE'.
C                           (VALID VALUES = <0, 0, 1, 2, 3, 4)
C
C          (L*4)  LSETX   = .TRUE.  => SET UP SPLINE PARAMETERS RELATING
C                                      TO 'XIN' AXIS.
C                           .FALSE. => DO NOT SET UP SPLINE PARAMETERS
C                                      RELATING TO 'XIN' AXIS.
C                                      (I.E. THEY WERE SET IN A PREVIOUS
C                                            CALL )
C                           (VALUE SET TO .FALSE. BY 'XXSPLE')
C
C          (R*8)  R8FUN1  = FUNCTION - (SEE ROUTINES SECTION BELOW)
C
C          (R*8)  XIN()   = 1) LOG( DATA FILE ELECTRON DENSITIES    )
C                           2) LOG( DATA FILE ELECTRON TEMPERATURES )
C          (R*8)  YIN()   = LOG( DATA FILE IONIZATIONS/PHOTON )
C          (R*8)  XOUT()  = 1) LOG( SCALED USER ENTERED ELECTRON DENS. )
C                           2) LOG( SCALED USER ENTERED ELECTRON TEMPS.)
C          (R*8)  YOUT()  = LOG( OUTPUT GENERATED IONIZATIONS/PHOTON )
C          (R*8)  YPASS(,)= LOG( IONIZATIONS/PHOTON) INTERMEDIATE ARRAY
C                           WHICH   STORES   INTERPOLATED/EXTRAPOLATED
C                           VALUES  BETWEEN  THE  TWO  SPLINE SECTIONS.
C                           SECTIONS.
C          (R*8)  DF()    = SPLINE INTERPOLATED DERIVATIVES
C
C
C NOTE:
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
```

```
C          ------------------------------------------------------------
C          XXSPLE     ADAS      SPLINE SUBROUTINE (EXTENDED DIAGNOSTICS)
C          R8FUN1     ADAS      REAL*8 FUNCTION: ( X -> X )
C
C AUTHOR:  H. P. SUMMERS
C          K1/1/57
C          JET EXT. 4941
C
C DATE:    11/10/91
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 28-02-95
C MODIFIED: LALIT JALOTA (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST VERSION
C
C VERSION: 1.2                          DATE: 15-04-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - INCREASED PARAMETER NIN 24 -> 26
C                 (CURRENT JET VALUE 35, BUT KEPT TO 26 FOR PRESENT TO
C                  PREVENT CHANGES TO IDL ROUTINES).
C
C VERSION: 1.3                          DATE: 22-04-96
C MODIFIED: TIM HAMMOND
C - INCREASED NIN TO 35 IN LINE WITH JET.
C
C VERSION: 1.4                          DATE: 13-10-97
C MODIFIED: Martin O'Mullane
C               - INCREASED PARAMETER NOUT 20 -> 35.
C
C VERSION: 1.5                          DATE: 20-09-99
C MODIFIED: RICHARD MARTIN
C               - INCREASED NOUT 35 -> 100 ; NIN 35 -> 100
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IDA,         ITA,         ITVAL,       NDDIM
      INTEGER             NTDIM
      LOGICAL             LDRNG(ITVAL),            LTRNG(ITVAL)
      REAL*8              DIN(ITVAL),  PEC(NTDIM,NDDIM)
      REAL*8              PECA(ITVAL), TEDA(IDA),   TETA(ITA)
      REAL*8              TEVA(ITVAL)
```

6.14 e3titl: Subroutine e3titl from library adas5xx

```
      SUBROUTINE E3TITL( IBSEL  , DSNAME ,
     &                   ESYM   , IZ     ,
     &                   CWAVEL , CINDM  ,
     &                   TITLX
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: E3TITL *********************
C
C  PURPOSE:  TO CREATE THE DESCRIPTIVE TITLE FOR SELECTED DATA-BLOCK.
C
C  CALLING PROGRAM: ADAS503/SPEC
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  IBSEL    = SELECTED DATA-BLOCK: INDEX
C  INPUT : (C*(*)) DSNAME   = FULL INPUT DATA SET NAME
C
C  INPUT : (C*2)  ESYM     = INPUT DATA SET: EMITTING ION ELEMENT SYMBOL
C  INPUT : (I*4)  IZ       = INPUT DATA SET: EMITTING ION CHARGE-STATE
C
C  INPUT : (C*10) CWAVEL   = SELECTED DATA-BLOCK: WAVELENGTH (ANGS.)
C  INPUT : (C*2)  CINDM    = SELECTED DATA-BLOCK: METASTABLE INDEX
C
C  OUTPUT: (C*120) TITLX    = SELECTED DATA-BLOCK: DESCRIPTIVE TITLE
C
C          (C*2)  C2       = GENERAL USE 2 BYTE  CHARACTER  STRING
C          (C*3)  C3       = GENERAL USE 3 BYTE  CHARACTER  STRING
C    (C*9)  PATH     = UNIX ENVIRONMENT OF CURRENT PATH=$ADASCENT
C    (I*4)  POS_NOW  = CURRENT POSITION IN TITLE STRING
C    (I*4)  LEN_NAME = LENGTH OF FILENAME
C          (I*4)  IFIRST   = POSITION OF FIRST CHARACTER OF DSNAME
C          (I*4)  ILAST    = POSITION OF LAST CHARACTER OF DSNAME
C
C ROUTINES:
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/37
C          JET EXT. 2520
C
C DATE:    01/05/91
C
C UPDATE:  L. JALOTA  13/3/95   MODIFIED FOR USE UNDER UNIX
C
C VERSION 1.7 TIM HAMMOND 13/10/95
C ADDED VARIABLE C3 TO ALLOW 3 DIGIT DATA BLOCKS
C               TO BE READ FROM THE VARIABLE IBSEL
C
C VERSION 1.8 TIM HAMMOND 08/11/95
C  - REMOVED SUPERFLUOUS VARIABLE TEMP_NAME
C
C VERSION 1.9 TIM HAMMOND 08/11/95
C  - REMOVED SUPERFLUOUS VARIABLE I4UNIT
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      CHARACTER*2         CINDM
      CHARACTER*10        CWAVEL
      CHARACTER*(*)       DSNAME
```

```
CHARACTER*2         ESYM
CHARACTER*120       TITLX
INTEGER             IBSEL,       IZ
```

6.15 e4chkb: Subroutine e4chkb from library adas5xx

```
      SUBROUTINE E4CHKB( IUNIT  , NBSEL  , IBSEL  ,
     &                   IZ0IN  , IZ0    ,
     &                   LOPEN  , IRCODE
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: E4CHKB *********************
C
C  PURPOSE: TO CHECK THE SELECTED BLOCK (IBSEL) OF DATA  EXISTS  IN  THE
C           INPUT DATA SET AND IF SO IT REPRESENTS THE ENTERED VALUES OF
C           'IZ0IN' (NUCLEAR CHARGE OF SELECTED IONISING ION ELEMENT).
C
C           IT ALSO CLOSES THE INPUT DATA SET ALLOCATION IF OPEN.
C
C  CALLING PROGRAM: SPZD
C
C  SUBROUTINE:
C
C  INPUT : (I*4)   IUNIT   = UNIT TO WHICH INPUT DATA SET IS ALLOCATED
C  INPUT : (I*4)   NBSEL   = TOTAL NUMBER OF DATA-BLOCKS READ FROM INPUT
C                            DATA SET.
C  INPUT : (I*4)   IBSEL   = INDEX OF DATA-BLOCK SELECTED FOR ANALYSIS
C
C  INPUT : (I*4)   IZ0IN   = REQUESTED: NUCLEAR CHARGE OF ELEMENT
C  INPUT : (I*4)   IZ0     = INPUT FILE: NUCLEAR CHARGE OF ELEMENT
C
C  I/O   : (L*4)   LOPEN   = INPUT : .TRUE.  => INPUT DATA SET OPEN.
C                                    .FALSE. => INPUT DATA SET CLOSED.
C                            OUTPUT: ALWAYS RETURNED AS .FALSE.
C  OUTPUT: (I*4)   IRCODE  = RETURN CODE FROM SUBROUTINE:
C                            0 => NO ERROR DETECTED.
C                            2 => DISCREPANCY BETWEEN REQUESTED ELEMENT
C                                 AND THAT IN INPUT DATA FILE.
C                            3 => SELECTED DATA-BLOCK  OUT OF RANGE  OR
C                                 DOES NOT EXIST.
C
C           (I*4)   I4UNIT  = FUNCTION (SEE ROUTINE SECTION BELOW)
C
C    (C*80)  DSNAME  = NAME OF DATA SET OPENED
C
C ROUTINES:
C           ROUTINE    SOURCE    BRIEF DESCRIPTION
C           ------------------------------------------------------------
C           E4FILE     ADAS      OPEN DATA SET FOR SELECTED ELEMENT
C           I4UNIT     ADAS      FETCH UNIT NUMBER FOR OUTPUT OF MESSAGES
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/37
C          JET EXT. 2620
C
C DATE:    06/06/91
C
C UPDATE:  23/04/93 - PE BRIDEN - ADAS91: ADDED I4UNIT FUNCTION TO WRITE
C                                         STATEMENTS FOR SCREEN MESSAGES
C
C UPDATE:  24/06/93 - PE BRIDEN - ADAS91: CHANGED I4UNIT(0)-> I4UNIT(-1)
C
C UPDATE:   1/11/94 - L. JALOTA - INCREASED SIZE OF DSNAME FOR UNIX.
C UNIX-IDL PORT:
```

```
C
C VERSION: 1.1                          DATE: 25-5-95
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - PUT UNDER SCCS CONTROL
C
C VERSION: 1.2                          DATE: 17-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - REPLACED SSZD WITH SPZD AND MINOR CHANGE TO FORMATS
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IBSEL,       IRCODE,      IUNIT,       IZ0
      INTEGER             IZ0IN,       NBSEL
      LOGICAL             LOPEN
```

6.16 e4data: Subroutine e4data from library adas5xx

```
       SUBROUTINE E4DATA( IUNIT  , DSNAME ,
     &                    NSTORE , NTDIM  ,
     &                    ESYM   , IZ0    ,
     &                    NBSEL  , ISELA  ,
     &                    IZ     , IZ1    ,
     &                    CIION  , CITYPE , CIINFO ,
     &                    ITA    ,
     &                    TETA   , PZD
     &                  )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: E4DATA *********************
C
C  PURPOSE:  TO  FETCH  DATA  FROM  INPUT RADIATED POWER COEFFICIENTS
C             OF AN ELEMENT AND ITS IONS.
C            (MEMBER STORED IN IONELEC.DATA - MEMBER PREFIX 'PZD#').
C
C  CALLING PROGRAM: ADAS504/SPZD
C
C  DATA:
C
C           UP TO 'NSTORE' SETS (DATA-BLOCKS) OF DATA MAY BE  READ FROM
C           THE FILE - EACH BLOCK FORMING A COMPLETE SET OF  RADIATED
C           POWER COEFFICIENT VALUES FOR GIVEN TEMPERATURES.
C           EACH DATA-BLOCK  IS  ANALYSED INDEPENDENTLY OF ANY  OTHER
C           DATA-BLOCK.
C
C           THE UNITS USED IN THE DATA FILE ARE TAKEN AS FOLLOWS:
C
C           TEMPERATURES        : EV
C           RATE COEFFT         : W CM**3
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  IUNIT    = UNIT TO WHICH INPUT FILE IS ALLOCATED.
C  INPUT : (C*80) DSNAME   = NAME OF DATA SET BEING READ
C
C  INPUT : (I*4)  NSTORE   = MAXIMUM NUMBER  OF  INPUT DATA-BLOCKS  THAT
C                            CAN BE STORED.
C  INPUT : (I*4)  NTDIM    = MAX NUMBER OF ELECTRON TEMPERATURES ALLOWED
C
C  OUTPUT: (C*2)  ESYM     = READ - IONISING ION - ELEMENT SYMBOL
C  OUTPUT: (I*4)  IZ0      = READ - IONISING ION - NUCLEAR CHARGE
C
C  OUTPUT: (I*4)  NBSEL    = NUMBER OF DATA-BLOCKS ACCEPTED & READ IN.
C  OUTPUT: (I*4)  ISELA()  = READ - DATA-SET DATA-BLOCK ENTRY INDICES
C                            DIMENSION: DATA-BLOCK INDEX
C
C  OUTPUT: (I*4)  IZ()     = READ - RADIATING ION CHARGE
C                                   ( SET TO -1 IF WHOLE ELEMENT)
C                            DIMENSION: DATA-BLOCK INDEX
C  OUTPUT: (I*4)  IZ1()    = READ - RADIATING ION CHARGE +1
C                                   ( SET TO 1 IF WHOLE ELEMENT)
C                            DIMENSION: DATA-BLOCK INDEX
C
C  OUTPUT: (C*5)  CIION()  = READ - RADIATING ION (AS <ESYM>+(IZ()> )
C                            DIMENSION: DATA-BLOCK INDEX
C  OUTPUT: (C*5)  CITYPE() = READ - RADIATION TYPE
C                            DIMENSION: DATA-BLOCK INDEX
```

```
C  OUTPUT: (C*20) CIINFO() = READ - INFORMATION STRING
C                            DIMENSION: DATA-BLOCK INDEX
C
C  OUTPUT: (I*4)  ITA()    = READ - NUMBER OF ELECTRON TEMPERATURES
C                            DIMENSION: DATA-BLOCK INDEX
C
C  OUTPUT: (R*8)  TETA(,)  = READ - ELECTRON TEMPERATURES (UNITS: eV)
C                            1st DIMENSION: ELECTRON TEMPERATURE INDEX
C                            2nd DIMENSION: DATA-BLOCK INDEX
C
C  OUTPUT: (R*8)  PZD(,)    =READ - FULL SET OF ZERO DENSITY RADIATED
C                                   POWER COEFFTS. (W CM**3)
C                            1st DIMENSION: ELECTRON TEMPERATURE INDEX
C                            2nd DIMENSION: DATA-BLOCK INDEX
C
C          (I*4)  I4EIZ0   = FUNCTION - (SEE ROUTINES SECTION BELOW)
C          (I*4)  I4FCTN   = FUNCTION - (SEE ROUTINES SECTION BELOW)
C          (I*4)  I4UNIT   = FUNCTION - (SEE ROUTINES SECTION BELOW)
C          (I*4)  IBLK     = ARRAY INDEX: DATA-BLOCK INDEX
C          (I*4)  ITT      = ARRAY INDEX: ELECTRON TEMPERATURE INDEX
C          (I*4)  NTNUM    = NUMBER OF ELECTRON TEMPERATURES FOR CURRENT
C                            DATA-BLOCK
C          (I*4)  IABT     = RETURN CODE FROM 'I4FCTN'
C          (I*4)  IPOS1    = GENERAL USE STRING INDEX VARIABLE
C          (I*4)  IPOS2    = GENERAL USE STRING INDEX VARIABLE
C
C
C          (L*4)  LBEND    = IDENTIFIES WHETHER THE LAST OF THE  INPUT
C                            DATA SUB-BLOCKS HAS BEEN LOCATED.
C                            (.TRUE. => END OF SUB-BLOCKS REACHED)
C
C          (C*1)  CSLASH   = '/' - DELIMITER FOR 'XXHKEY'
C          (C*2)  C2       = GENERAL USE TWO BYTE CHARACTER STRING
C          (C*4)  CKEY1    = '****'  - INPUT BLOCK HEADER KEY
C          (C*5)  CKEY2    = 'TYPE'  - INPUT BLOCK HEADER KEY
C          (C*5)  CKEY3    = 'INFO'  - INPUT BLOCK HEADER KEY
C          (C*4)  CKEY4    = 'ISEL'  - INPUT BLOCK HEADER KEY
C          (C*10) C10      = GENERAL USE TEN BYTE CHARACTER STRING
C          (C*80) C80      = GENERAL USE 80 BYTE  CHARACTER  STRING  FOR
C                            THE INPUT OF DATA-SET RECORDS.
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          XXHKEY     ADAS      OBTAIN KEY/RESPONSE STRINGS FROM TEXT
C          I4EIZ0     ADAS      INTEGER*4 FUNCTION    -
C                               RETURNS Z0 FOR GIVEN ELEMENT SYMBOL
C          I4FCTN     ADAS      INTEGER*4 FUNCTION    -
C                               CONVERT CHARACTER STRING TO INTEGER
C          I4UNIT     ADAS      INTEGER*4 FUNCTION    -
C                               FETCH UNIT NUMBER FOR OUTPUT OF MESSAGES
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/37
C          JET EXT. 2520
C
C DATE:    07/06/91
C
C UPDATE:  23/04/93 - PE BRIDEN - ADAS91: ADDED I4UNIT FUNCTION TO WRITE
C                                         STATEMENTS FOR SCREEN MESSAGES
C
```

```
C UPDATE:  24/05/93 - PE BRIDEN - ADAS91: CHANGED I4UNIT(0)-> I4UNIT(-1)
C
C UPDATE:   4/11/94 - L. JALOTA - INCREASED SIZE OF DSNAME TO RUN UNDER
C           UNIX
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 17-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST VERSION
C
C VERSION: 1.2                          DATE: 17-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)/HUGH SUMMERS
C               - MINOR MODIFICATIONS TO FORMAT STATEMENTS
C
C-----------------------------------------------------------------------
      CHARACTER*20        CIINFO(NSTORE)
      CHARACTER*5         CIION(NSTORE),            CITYPE(NSTORE)
      CHARACTER*80        DSNAME
      CHARACTER*2         ESYM
      INTEGER             ISELA(NSTORE),            ITA(NSTORE), IUNIT
      INTEGER             IZ(NSTORE),  IZ0,         IZ1(NSTORE), NBSEL
      INTEGER             NSTORE,      NTDIM
      REAL*8              PZD(NTDIM,NSTORE),        TETA(NTDIM,NSTORE)
```

6.17 e4file: Subroutine e4file from library adas5xx

```
      SUBROUTINE E4FILE( IUNIT , IZ0 , IRCODE , DSNAME )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: E4FILE *********************
C
C  PURPOSE: TO OPEN AN RADIATED POWER COEFFT 'IONELEC' DATA SET
C           BY DEFAULT, OR AN ALTERNATIVE DATA SET IF REQUIRED, FOR
C           RADIATING ELEMENT GIVEN BY IT NUCLEAR CHARGE 'IZ0'.
C           THIS WILL BE CONNECTED TO UNIT 'IUNIT'.
C
C  DATA SET OPENED: '$ADASUSER/DEFADF/<GROUP>/<TYPE>/
C <GROUP>_<EXT>#<ELEMENT SYMBOL>'
C
C  CALLING PROGRAM: SPZD
C
C  SUBROUTINE:
C
C  INPUT : (I*4)   IUNIT   = UNIT TO WHICH DATA SET WILL BE CONNECTED
C  INPUT : (I*4)   IZ0     = NUCLEAR CHARGE OF ELEMENT REQUESTED
C
C  OUTPUT: (I*4)   IRCODE  = RETURN CODE FROM SUBROUTINE:
C                            0 => DATA SET SUCCESSFULLY CONNECTED
C                            1 => REQUESTED DATA  SET  MEMBER  DOES  NOT
C                                 EXISTS - DATA SET NOT CONNECTED.
C                            9 => REQUESTED DATA  SET  EXISTS BUT CANNOT
C                                 BE OPENED.
C  OUTPUT: (C*80)  DSNAME  = NAME OF OPENED DATA SET UNDER UNIX
C
C          (I*4)   IDLEN   = LENGTH, IN BYTES, OF FIXED 'DSNAME' PREFIX
C          (I*4)   LENF1   = FIRST NON-BLANK CHR OF 'DSNAME' GROUP PART
C          (I*4)   LENF2   = LAST  NON-BLANK CHR OF 'DSNAME' GROUP PART
C          (I*4)   LENF3   = FIRST NON-BLANK CHR OF 'DSNAME' TYPE PART
C          (I*4)   LENF4   = LAST  NON-BLANK CHR OF 'DSNAME' TYPE PART
C    (I*4)   LENF5   = FIRST NON-BLANK CHR OF 'DSNAME' USERID PART
C          (I*4)   LENF6   = LAST  NON-BLANK CHR OF 'DSNAME' USERID PART
C          (I*4)   LENF7   = FIRST NON-BLANK CHR OF 'DSNAME' USREXT PART
C          (I*4)   LENF8   = LAST  NON-BLANK CHR OF 'DSNAME' USREXT PART
CA         (C*1)   HASH    = '#'   IF NON-BLANK EXT, ELSE ' '.
C          (C*2)   XFESYM  = FUNCTION - (SEE ROUTINES SECTION BELOW)
C          (C*2)   ESYM    = ELEMENT SYMBOL FOR NUCLEAR CHARGE 'IZ0'
C          (C*3)   USREXT  = ADAS SOURCE DATA FILE EXTENSION
C          (C*80)  USERID  = ADAS SOURCE DATA USER ID
C          (C*8)   USRGRP  = ADAS SOURCE DATA GROUPNAME
CA         (C*80)  USRTYP  = SUB-DIRECTORY NAME (OPTIONAL)
CA         (C*5)   DEFADF  = DEFAULT ADF DIRECTORY FOR THIS ROUTINE,I.E.
CA                           ADF19
C
C          (L*4)   LEXIST  = .TRUE.  => REQUESTED  DATA  SET  EXISTS.
C                           .FALSE. => REQUESTED  DATA  SET  DOES  NOT
C                                      EXIST.
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          XXUID      ADAS      FETCHES/SETS ADAS SOURCE DATA USER ID
C          XXSPZD     ADAS      FETCHES/SETS ADAS SOURCE DATA FILENAME
C                               AND FILE EXTENSION
C          XFESYM     ADAS      CHARACTER*2 FUNCTION -
C                               GATHERS ELEMENT SYMBOL FOR NUC. CHARGE
```

```
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/37
C          JET EXT. 5023
C
C DATE:    06/06/91
C UPDATE:  31/07/91 - PE BRIDEN: ADDED "ACTION='READ'" TO OPEN STATEMENT
C UPDATE:  08/08/91 - PE BRIDEN: ADDED "IRCODE=9" => OPEN ERROR
C UPDATE:  10/03/93 - PE BRIDEN: ADDED CALL TO XXUID AND USERID VARIABLE
C UPDATE:   2/09/93 - HPS      : ADDED CALL TO XXSPZD AND USRGRP, USRTYP
C                                  AND USREXT NAMES
C                                - NOW ALLOWS ANY INPUT DATASET FILENAME
C                                  AND EXTENSION
C                                - NOW ALLOWS ANY INPUT DATASET USER ID.
C UPDATE:  23/11/93 - PEB      : CORRECT ERROR - A '.' HAD MISTAKENLY
C                                BEEN PLACED BEFORE THE MEMBER NAME IN
C                                VARIABLE DSNAME.
C UPDATE:   4/11/94 - L.JALOTA : MODIFIED FOR UNIX,INCREASED NAME LENGTHS
C UPDATE:   7/11/94 - L.JALOTA : ADDED DEFADF.
C UPDATE:  21/11/94 - L.JALOTA : CORRECTED SIZE OF USRTYP.
C
C UPDATE:  24/03/95 - HPS      : INTRODUCED HASH TO ELIMINATE # IN FILE IF
C                                THERE IS NO EXTENSION PART OF THE FILE NAME
C                                ALTER LOGIC TO ALLOW USRTYP, USREXT TO BE A
C                                SINGLE CHARACTER.
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      CHARACTER*80        DSNAME
      INTEGER             IRCODE,      IUNIT,       IZ0
```

6.18 e4spln: Subroutine e4spln from library adas5xx

```
      SUBROUTINE E4SPLN( ITA   , ITVAL   ,
     &                   TETA   , TEVA    ,
     &                   PZD    , PZDA    ,
     &                   LTRNG
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: E4SPLN *********************
C
C  PURPOSE:
C          PERFORMS CUBIC SPLINE ON LOG(TEMPERATURE <EV> ) VERSUS
C          LOG(SCALED RADIATED POWER COEFFICIENTS).
C          INPUT DATA FOR A GIVEN IONZING ION COMBINATION DATA-BLOCK.
C
C          USING  ONE-WAY SPLINES  IT  CALCULATES  THE  RADIATED POWER
C          COEFFICIENT  FOR  'ITVAL'  ELECTRON  TEMPERATURE VALUES  FROM
C          THE LIST OF ELECTRON TEMPERATURES READ IN FROM THE INPUT FILE
C
C          IF A  VALUE  CANNOT  BE  INTERPOLATED  USING  SPLINES  IT  IS
C          EXTRAPOLATED VIA 'XXSPLE'. (SEE NOTES BELOW).
C
C  CALLING PROGRAM: ADAS504/SPZD
C
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  ITA     = INPUT DATA FILE: NUMBER OF ELECTRON TEMPERA-
C                           TURES READ FOR THE DATA-BLOCK BEING ASSESSED
C  INPUT : (I*4)  ITVAL   = NUMBER OF ISPF ENTERED ELECTRON TEMPERATURE
C                           VALUES FOR  WHICH  IOINIZATION RATE COEFFTS
C                           ARE REQUIRED FOR TABULAR/GRAPHICAL OUTPUT.
C
C  INPUT : (R*8)  TETA()  = INPUT DATA FILE: ELECTRON TEMPERATURES (EV)
C                           FOR THE DATA-BLOCK BEING ASSESSED.
C                           DIMENSION: ELECTRON TEMPERATURE INDEX
C  INPUT : (R*8)  TEVA()  = USER ENTERED: ELECTRON TEMPERATURES (EV)
C                           DIMENSION: TEMPERATURE/DENSITY PAIR INDEX
C
C  INPUT : (R*8)  PZD()    =INPUT DATA FILE: FULL SET OF ZERO DENSITY
C                           RADIATED POWER COEFFTS FOR THE DATA-BLOCK
C                           BEING ANALYSED.
C                           1ST DIMENSION: ELECTRON TEMPERATURE INDEX
C  OUTPUT: (R*8)  PZDA()  = SPLINE INTERPOLATED OR  EXTRAPOLATED  ZERO
C                           DENSITY RADIATED POWER COEFFICIENTS FOR
C                           THE USER ENTERED ELECTRON TEMPERATURES.
C                           DIMENSION: ELECTRON TEMPERATURE INDEX
C
C  OUTPUT: (L*4)  LTRNG()=  .TRUE.  => OUTPUT 'PZDA()' VALUE WAS INTER-
C                                      POLATED  FOR  THE  USER  ENTERED
C                                      ELECTRON TEMPERATURE 'TEVA()'.
C                           .FALSE. => OUTPUT 'PZDA()' VALUE WAS EXTRA-
C                                      POLATED  FOR  THE  USER  ENTERED
C                                      ELECTRON TEMPERATURE 'TEVA()'.
C                           DIMENSION: ELECTRON TEMPERATURE INDEX
C
C          (I*4)  NIN     = PARAMETER = MAX. NO. OF INPUT  TEMPERATURE
C                                       VALUES. MUST BE >= 'ITA'
C          (I*4)  NOUT    = PARAMETER = MAX. NO. OF OUTPUT TEMPERATURE
C                                       PAIRS.  MUST BE >= 'ITVAL'
```

```
C          (I*4)  L1      = PARAMETER = 1
C
C          (R*8)  BCONST  = PARAMETER = 1/(SCALED BOLTZMANN'S CONSTANT)
C
C          (I*4)  IET     = ARRAY SUBSCRIPT USED INPUT  FILE  ELECTRON
C                           TEMPERATURES.
C          (I*4)  IT      = ARRAY  SUBSCRIPT  USED  FOR  USER  ENTERED
C                           TEMPERATURE VALUES.
C          (I*4)  IOPT    = DEFINES THE BOUNDARY DERIVATIVES FOR THE
C                           SPLINE ROUTINE 'XXSPLE', SEE 'XXSPLE'.
C                           (VALID VALUES = <0, 0, 1, 2, 3, 4)
C
C          (L*4)  LSETX   = .TRUE.  => SET UP SPLINE PARAMETERS RELATING
C                                      TO 'XIN' AXIS.
C                           .FALSE. => DO NOT SET UP SPLINE PARAMETERS
C                                      RELATING TO 'XIN' AXIS.
C                                      (I.E. THEY WERE SET IN A PREVIOUS
C                                            CALL )
C                           (VALUE SET TO .FALSE. BY 'XXSPLE')
C
C          (R*8)  R8FUN1  = FUNCTION - (SEE ROUTINES SECTION BELOW)
C
C          (R*8)  XIN()   = LOG( DATA FILE ELECTRON TEMPERATURES )
C          (R*8)  YIN()   = LOG( DATA FILE SCALED ION. RATE COEFFTS.)
C          (R*8)  XOUT()  = LOG( USER ENTERED ELECTRON TEMPS.)
C          (R*8)  YOUT()  = LOG( OUTPUT GENERATED SCALED ION. RATE COEF)
C          (R*8)  DF()    = SPLINE INTERPOLATED DERIVATIVES
C
C
C NOTE:
C
C          ONE-DIMENSIONAL SPLINE CARRIED OUT BY THIS SUBROUTINE:
C
C           LOG( Pzd )  VS. LOG( Te )
C
C           Te      = electron temperature (units: eV)
C           Pzd     = zero density radiated power coefficient
C                                     (units: W cm**3)
C
C           Extrapolation criteria:
C
C           Low  Te: zero gradient  extrapolation (i.e. DY(1)  = 0.0)
C           High Te: zero curvature extrapolation (i.e. DDY(N) = 0.0)
C
C           (These criteria are met by calling XXSPLE with IOPT=4)
C
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          XXSPLE     ADAS      SPLINE SUBROUTINE (EXTENDED DIAGNOSTICS)
C          R8FUN1     ADAS      REAL*8 FUNCTION: ( X -> X )
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/37
C          JET EXT. 2620
C
C DATE:    07/06/91
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 17-1-96
```

```
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST VERSION
C
C VERSION: 1.2                          DATE: 17-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - REMOVED SUPERFLUOUS VARIABLES AND TIDIED COMMENTS
C                 SLIGHTLY
C
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             ITA,         ITVAL
      LOGICAL             LTRNG(ITVAL)
      REAL*8              PZD(ITA),    PZDA(ITVAL), TETA(ITA)
      REAL*8              TEVA(ITVAL)
```

6.19 e4titl: Subroutine e4titl from library adas5xx

```
      SUBROUTINE E4TITL( IBSEL  , DSNAME ,
     &                   CIION  , CITYPE , CIINFO ,
     &                   TITLX
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: E4TITL *********************
C
C  PURPOSE:  TO CREATE THE DESCRIPTIVE TITLE FOR SELECTED DATA-BLOCK.
C
C  CALLING PROGRAM: ADAS504/SPZD
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  IBSEL    = SELECTED DATA-BLOCK: INDEX
C  INPUT : (C*80) DSNAME   = DATA FILE NAME UNDER UNIX INCLUDING PATH
C
C  INPUT : (C*5)  CIION    = SELECTED DATA-BLOCK: RADIATING ION
C  INPUT : (C*5)  CITYPE   = SELECTED DATA-BLOCK: RADIATION TYPE
C  INPUT : (C*20) CIINFO   = SELECTED DATA-BLOCK: INFORMATION STRING
C
C  OUTPUT: (C*120) TITLX    = SELECTED DATA-BLOCK: DESCRIPTIVE TITLE
C
C          (C*2)  C2       = GENERAL USE 2 BYTE  CHARACTER  STRING
C    (I*4)  POS_NOW  = CURRENT POSITION IN TITLE STRING
C    (I*4)  LEN_NAME = LENGTH OF FILENAME
C          (I*4)  IFIRST   = POSITION OF FIRST CHARACTER IN FILENAME
C          (I*4)  ILAST    = POSITION OF LAST CHARACTER IN FILENAME
C
C
C ROUTINES:
CA         XXSLEN          = UTILITY ROUTINE WHICH FINDS FIRST AND LAST
CA                           NON-BLANK CHARACTERS IN A STRING.
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/37
C          JET EXT. 2620
C
C DATE:    07/06/91
C
C UNIX-IDL PORT:
C
C UPDATE:  04/11/94 - L. JALOTA: MODIFIED FOR USE UNDER UNIX
C
C UPDATE:  26/03/95 - HPS      : CORRECTED CIION AND CITYP TO C*5
C
C VERSION: 1.1                          DATE: 25-5-95
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - PUT UNDER SCCS CONTROL
C
C UPDATE:  HP SUMMERS  20/12/95  - UNIX-IDL PORT
C
C VERSION: 1.2                          DATE: 17-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - MADE CHANGES IN LINE WITH THOSE MADE BY HUGH SUMMERS
C                 AT STRATHCLYDE - INCLUDING TIDYING UP COMMENTS AND
C                 HOW THE TITLE IS CONSTRUCTED
C
C-----------------------------------------------------------------------
```

```
C-----------------------------------------------------------------------
      CHARACTER*20        CIINFO
      CHARACTER*5         CIION,       CITYPE
      CHARACTER*80        DSNAME
      CHARACTER*120       TITLX
      INTEGER             IBSEL
```

6.20 e5data: Subroutine e5data from library adas5xx

```
       SUBROUTINE E5DATA( IUNIT  , DSNAME ,
     &                    NSTORE , NTRDIM , NTDDIM ,
     &                    NBSEL  , ISELA  ,
     &                    IRZ0   , IRZ1   , IDZ0   ,
     &                    LEQUA  ,
     &                    CDONOR , CRECVR , CFSTAT ,
     &                    AMSRA  , AMSDA  ,
     &                    ITRA   , ITDA   ,
     &                    TFRA   , TFDA   ,
     &                    QFTEQA , QFTCXA
     &                  )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: E5DATA *********************
C
C  PURPOSE:  TO  FETCH  DATA  FROM  INPUT THERMAL TOTAL CHARGER TRANSFER
C            RATE COEFFICIENT FILE FOR GIVN RECEIVER ION ELEMENT.
C            (MEMBER STORED IN IONATOM.DATA - MEMBER PREFIX 'TCX#').
C
C  CALLING PROGRAM: ADAS505/SQTCX
C
C  DATA:
C
C           UP TO 'NSTORE' SETS (DATA-BLOCKS) OF DATA MAY BE  READ FROM
C           THE FILE - EACH  BLOCK  FORMING  A  COMPLETE  SET OF  RATE-
C           COEFFICIENTS FOR A GIVEN  RECEIVER/DONOR COMBINATION.  EACH
C           DATA-BLOCK  IS  ANALYSED INDEPENDENTLY OF ANY  OTHER  DATA-
C           BLOCK.
C
C           THE UNITS USED IN THE DATA FILE ARE TAKEN AS FOLLOWS:
C
C           TEMPERATURES        : EV
C           RATE COEFFICIENTS   : CM**3 SEC-1
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  IUNIT    = UNIT TO WHICH INPUT FILE IS ALLOCATED.
C  INPUT : (C*44) DSNAME   = MVS DATA SET NAME OF DATA SET BEING READ
C
C  INPUT : (I*4)  NSTORE   = MAXIMUM NUMBER  OF  INPUT DATA-BLOCKS  THAT
C                            CAN BE STORED.
C  INPUT : (I*4)  NTRDIM   = MAX NUMBER OF RECEIVER TEMPERATURES ALLOWED
C  INPUT : (I*4)  NTDDIM   = MAX NUMBER OF DONOR    TEMPERATURES ALLOWED
C
C  OUTPUT: (I*4)  NBSEL    = NUMBER OF DATA-BLOCKS ACCEPTED & READ IN.
C  OUTPUT: (I*4)  ISELA()  = READ - DATA-SET DATA-BLOCK ENTRY INDICES
C                            DIMENSION: DATA-BLOCK INDEX
C
C  OUTPUT: (I*4)  IRZ0()   = NUCLEAR CHARGE OF RECEIVING IMPURITY ION -
C                            READ FROM SELECTED DATA-BLOCK.
C                            DIMENSION: DATA-BLOCK INDEX.
C  OUTPUT: (I*4)  IRZ1()   = INITIAL CHARGE OF RECEIVER -
C                            READ FROM SELECTED DATA-BLOCK.
C                            DIMENSION: DATA-BLOCK INDEX.
C  OUTPUT: (I*4)  IDZ0()   = NUCLEAR CHARGE OF NEUTRAL DONOR -
C                            READ FROM SELECTED DATA-BLOCK.
C                            DIMENSION: DATA-BLOCK INDEX.
C
C  OUTPUT: (L*4)  LEQUA()  = READ - DATA SET ENTRY FORMAT
```

```
C                                       .TRUE.  => DATA  SET CONTAINS  EQUAL
C                                                  TEMPERATURE COEFFICIENT.
C                                       .FALSE. => DATA SET DOES NOT CONTAIN
C                                                  EQUAL TEMPERATURE COEFFT.
C                                 DIMENSION: DATA-BLOCK INDEX
C  OUTPUT: (C*9)  CDONOR() = READ - DONOR ION IDENTIFICATION
C                                 DIMENSION: DATA-BLOCK INDEX
C  OUTPUT: (C*9)  CRECVR() = READ - RECEIVER ION IDENTIFICATION
C                                 DIMENSION: DATA-BLOCK INDEX
C  OUTPUT: (C*10) CFSTAT() = READ - FINAL STATE SPECIFICATION
C                                 DIMENSION: DATA-BLOCK INDEX
C  OUTPUT: (R*8)  AMSRA()  = READ - RECEIVER ATOMIC MASS
C                                 DIMENSION: DATA-BLOCK INDEX
C  OUTPUT: (R*8)  AMSDA()  = READ - DONOR    ATOMIC MASS
C                                 DIMENSION: DATA-BLOCK INDEX
C
C  OUTPUT: (I*4)  ITRA()   = READ - NUMBER OF RECEIVER TEMPERATURES
C                                 DIMENSION: DATA-BLOCK INDEX
C  OUTPUT: (I*4)  ITDA()   = READ - NUMBER OF DONOR    TEMPERATURES
C                                 DIMENSION: DATA-BLOCK INDEX
C
C  OUTPUT: (R*8)  TFRA(,)  = READ - RECEIVER TEMPERATURES (UNITS: EV)
C                                 1ST DIMENSION: RECEIVER TEMPERATURE INDEX
C                                 2ND DIMENSION: DATA-BLOCK INDEX
C  OUTPUT: (R*8)  TFDA(,)  = READ - DONOR    TEMPERATURES (UNITS: EV)
C                                 1ST DIMENSION: DONOR    TEMPERATURE INDEX
C                                 2ND DIMENSION: DATA-BLOCK INDEX
C
C  OUTPUT: (R*8)  QFTEQA(,)= READ - EQUAL TEMPERATURE RATE-COEFFICIENTS
C                                        (UNITS: CM**3 SEC-1)
C                                 1ST DIMENSION: RECEIVER TEMPERATURE INDEX
C                                 2ND DIMENSION: DATA-BLOCK INDEX
C  OUTPUT: (R*8)  QFTCXA(,,)=READ - FULL SET OF RATE-COEFFICIENTS
C                                        (UNITS: CM**3 SEC-1)
C                                 1ST DIMENSION: DONOR    TEMPERATURE INDEX
C                                 2ND DIMENSION: RECEIVER TEMPERATURE INDEX
C                                 3RD DIMENSION: DATA-BLOCK INDEX
C
C          (C*2)  CEQUAL   = PARAMETER = 'EQ'
C
C          (I*4)  I4EIZ0   = FUNCTION - (SEE ROUTINES SECTION BELOW)
C          (I*4)  I4FCTN   = FUNCTION - (SEE ROUTINES SECTION BELOW)
C          (I*4)  I4UNIT   = FUNCTION - (SEE ROUTINE SECTION BELOW)
C          (I*4)  IBLK     = ARRAY INDEX: DATA-BLOCK INDEX
C          (I*4)  ITR      = ARRAY INDEX: RECEIVER TEMPERATURE INDEX
C          (I*4)  ITD      = ARRAY INDEX: DONOR    TEMPERATURE INDEX
C          (I*4)  NTRNUM   = NUMBER OF RECEIVER TEMPERATURES FOR CURRENT
C                            DATA-BLOCK
C          (I*4)  NTDNUM   = NUMBER OF DONOR    TEMPERATURES FOR CURRENT
C                            DATA-BLOCK
C    (I*7)  N7     = MIN(7,NDTNUM) REQUIRED TO HANDLE > 7 DONOR TEMPS
C          (I*4)  IABT     = RETURN CODE FROM 'I4FCTN'
C
C          (L*4)  LBEND    = IDENTIFIES WHETHER THE LAST OF THE  INPUT
C                            DATA SUB-BLOCKS HAS BEEN LOCATED.
C                            (.TRUE. => END OF SUB-BLOCKS REACHED)
C
C          (C*10) IONNAM   = READ - DONOR/RECEIVER DESIGNATION STRING
C
C          (C*1)  CSLASH   = '/' - DELIMITER FOR 'XXHKEY'
C          (C*2)  C2       = GENERAL USE TWO BYTE CHARACTER STRING
```

```
C          (C*3)  CKEY1    = 'AMD'    - INPUT BLOCK HEADER KEY
C          (C*3)  CKEY2    = 'AMR'    - INPUT BLOCK HEADER KEY
C          (C*3)  CKEY3    = 'FST'    - INPUT BLOCK HEADER KEY
C          (C*4)  CKEY4    = 'ISEL'  - INPUT BLOCK HEADER KEY
C          (C*80) C80      = GENERAL USE 80 BYTE  CHARACTER  STRING  FOR
C                            THE INPUT OF DATA-SET RECORDS.
C          (C*80) C80      = GENERAL USE 80 BYTE  CHARACTER  STRING  FOR
C                            THE INPUT OF DATA-SET RECORDS.
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          XXHKEY     ADAS      OBTAIN KEY/RESPONSE STRINGS FROM TEXT
C          I4EIZ0     ADAS      INTEGER*4 FUNCTION    -
C                               RETURNS Z0 FOR GIVEN ELEMENT SYMBOL
C          I4FCTN     ADAS      INTEGER*4 FUNCTION    -
C                               CONVERT CHARACTER STRING TO INTEGER
C          R8FCTN     ADAS      REAL*8 FUNCTION    -
C                               CONVERT CHARACTER STRING TO REAL
C          I4UNIT     ADAS      INTEGER*4 FUNCTION    -
C                               FETCH UNIT NUMBER FOR OUTPUT OF MESSAGES
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 4569
C
C DATE:    20/02/91
C
C UPDATE:  23/04/93 - PE BRIDEN - ADAS91: ADDED I4UNIT FUNCTION TO WRITE
C                                         STATEMENTS FOR SCREEN MESSAGES
C
C UPDATE:  24/05/93 - PE BRIDEN - ADAS91: CHANGED I4UNIT(0)-> I4UNIT(-1)
C
C UPDATE:  15/12/95 - HP SUMMERS- ADAS91: MODIFIED INFOMATION STRING USE
C
C UNIX-IDL PORT:
C
C AUTHOR:  WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C          NO CHANGES FROM IBM VERSION
C
C DATE:    20TH MARCH 1996
C
C VERSION: 1.1                  DATE: 20-03-96
C MODIFIED: WILLIAM OSBORN
C           - FIRST VERSION
C
C VERSION: 1.2                  DATE: 10-04-96
C MODIFIED: WILLIAM OSBORN
C           - REMOVED REDUNDANT VARIABLE
C
C VERSION: 1.3 DATE: 19-09-97
C MODIFIED: MARTIN O'MULLANE
C     - MODIFIED TO ALLOW DATASETS WITH GREATER THAN 7 DONOR
C       TEMPERATURES TO BE USED.
C
C-----------------------------------------------------------------------
      CHARACTER*9         CDONOR(NSTORE)
      CHARACTER*10        CFSTAT(NSTORE)
      CHARACTER*9         CRECVR(NSTORE)
      CHARACTER*44        DSNAME
      INTEGER             IDZ0(NSTORE),             IRZ0(NSTORE)
```

```
INTEGER             IRZ1(NSTORE),              ISELA(NSTORE)
INTEGER             ITDA(NSTORE),              ITRA(NSTORE)
INTEGER             IUNIT,       NBSEL,        NSTORE,      NTDDIM
INTEGER             NTRDIM
LOGICAL             LEQUA(NSTORE)
REAL*8              AMSDA(NSTORE),             AMSRA(NSTORE)
REAL*8              QFTCXA(NTDDIM,NTRDIM,NSTORE)
REAL*8              QFTEQA(NTRDIM,NSTORE),     TFDA(NTDDIM,NSTORE)
REAL*8              TFRA(NTRDIM,NSTORE)
```

6.21 e5spln: Subroutine e5spln from library adas5xx

```
      SUBROUTINE E5SPLN( NTRDIM , NTDDIM ,
     &                   ITRA   , ITDA   , ITVAL   ,
     &                   AMSRA  , AMSDA  , RMASS   , DMASS  ,
     &                   TFRA   , TFDA   , TREVA   , TDEVA  ,
     &                   QFTCXA ,          QTCXA   ,
     &                                     LTRRNG  , LTDRNG
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: E5SPLN *********************
C
C  PURPOSE:
C          PERFORMS CUBIC SPLINE ON LOG(TEMP.) VERSUS LOG(RATE-COEFT)
C          INPUT DATA FOR A GIVEN DONOR/RECEIVER DATA-BLOCK.
C
C          USING TWO-WAY SPLINES IT CALCULATES THE RATE-COEFFICIENTS
C          FOR 'ITVAL' PAIRS OF RECEIVER/DONOR TEMPERATURES FROM THE
C          TWO-DIMENSIONAL TABLE OF RECEIVER/DONOR TEMPERATURES READ
C          IN FROM THE INPUT FILE. IF A VALUE CANNOT BE INTERPOLATED
C          USING SPLINES IT IS EXPLICITLY EXTRAPOLATED.
C
C  CALLING PROGRAM: ADAS505/SQTCX
C
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  NTRDIM  = MAX NUMBER OF RECEIVER TEMPERATURES ALLOWED
C  INPUT : (I*4)  NTDDIM  = MAX NUMBER OF DONOR    TEMPERATURES ALLOWED
C
C  INPUT : (I*4)  ITRA    = INPUT DATA FILE: NUMBER OF RECEIVER TEMPERA-
C                           TURES READ FOR THE DATA-BLOCK BEING ASSESSED
C  INPUT : (I*4)  ITDA    = INPUT DATA FILE: NUMBER OF DONOR    TEMPERA-
C                           TURES READ FOR THE DATA-BLOCK BEING ASSESSED
C  INPUT : (I*4)  ITVAL   = NUMBER OF ISPF ENTERED RECEIVER/DONOR  TEMP-
C                           ERATURE PAIRS  FOR  WHICH  RATE-COEFFICIENTS
C                           ARE REQUIRED FOR TABULAR/GRAPHICAL OUTPUT.
C
C  INPUT : (R*8)  AMSRA   = INPUT DATA FILE: RECEIVER ATOMIC MASS FOR
C                           THE DATA-BLOCK BEING ASSESSED.
C  INPUT : (R*8)  AMSDA   = INPUT DATA FILE: DONOR    ATOMIC MASS FOR
C                           THE DATA-BLOCK BEING ASSESSED.
C  INPUT : (R*8)  RMASS   = USER ENTERED: RECEIVER ISOTOPIC ATOMIC MASS
C  INPUT : (R*8)  DMASS   = USER ENTERED: DONOR    ISOTOPIC ATOMIC MASS
C
C  INPUT : (R*8)  TFRA()  = INPUT DATA FILE: RECEIVER TEMPERATURES (EV)
C                           FOR THE DATA-BLOCK BEING ASSESSED.
C                           DIMENSION: RECEIVER TEMPERATURE INDEX
C  INPUT : (R*8)  TFDA()  = INPUT DATA FILE: DONOR    TEMPERATURES (EV)
C                           FOR THE DATA-BLOCK BEING ASSESSED.
C                           DIMENSION: DONOR    TEMPERATURE INDEX
C  INPUT : (R*8)  TREVA() = USER ENTERED: RECEIVER TEMPERATURES (EV)
C                           DIMENSION: RECEIVER/DONOR TEMP. PAIR INDEX
C  INPUT : (R*8)  TDEVA() = USER ENTERED: DONOR    TEMPERATURES (EV)
C                           DIMENSION: RECEIVER/DONOR TEMP. PAIR INDEX
C
C
C  INPUT : (R*8)  QFTCXA(,)=INPUT DATA FILE: FULL SET OF RATE-COEFFTS.
C                           (UNITS: CM**3/SEC) FOR THE DATA-BLOCK BEING
```

```
C                            ANALYSED.
C                            1ST DIMENSION: DONOR    TEMPERATURE INDEX
C                            2ND DIMENSION: RECEIVER TEMPERATURE INDEX
C  OUTPUT: (R*8)  QTCXA() = SPLINE INTERPOLATED OR  EXTRAPOLATED  RATE-
C                            COEFFICIENTS FOR THE USER ENTERED RECEIVER/
C                            DONOR TEMPERATURE PAIRS (UNITS: CM**3/SEC)
C                            DIMENSION: RECEIVER/DONOR TEMP. PAIR INDEX
C
C  OUTPUT: (L*4)  LTRRNG()= .TRUE.  => OUTPUT 'QTCXA()' VALUE WAS INTER-
C                                      POLATED  FOR  THE  USER  ENTERED
C                                      RECEIVER TEMPERATURE 'TREVA()'.
C                            .FALSE. => OUTPUT 'QTCXA()' VALUE WAS EXTRA-
C                                      POLATED  FOR  THE  USER  ENTERED
C                                      RECEIVER TEMPERATURE 'TREVA()'.
C                            DIMENSION: RECEIVER/DONOR TEMP. PAIR INDEX
C
C  OUTPUT: (L*4)  LTDRNG()= .TRUE.  => OUTPUT 'QTCXA()' VALUE WAS INTER-
C                                      POLATED  FOR  THE  USER  ENTERED
C                                      DONOR TEMPERATURE 'TDEVA()'.
C                            .FALSE. => OUTPUT 'QTCXA()' VALUE WAS EXTRA-
C                                      POLATED  FOR  THE  USER  ENTERED
C                                      DONOR TEMPERATURE 'TDEVA()'.
C
C          (I*4)  NIN     = PARAMETER = MAX. NO. OF INPUT  TEMPERATURE
C                                       VALUES. MUST BE >= 'ITRA'&'ITDA'
C          (I*4)  NOUT    = PARAMETER = MAX. NO. OF OUTPUT TEMPERATURE
C                                       PAIRS.  MUST BE >= 'ITVAL'
C          (I*4)  L1      = PARAMETER = 1
C
C          (I*4)  ITD     = ARRAY SUBSCRIPT USED INPUT  FILE  DONOR
C                           TEMPERATURES.
C          (I*4)  ITR     = ARRAY SUBSCRIPT USED INPUT  FILE  RECEIVER
C                           TEMPERATURES.
C          (I*4)  IT      = ARRAY  SUBSCRIPT  USED  FOR  USER  ENTERED
C                           TEMPERATURE PAIRS .
C          (I*4)  IOPT    = DEFINES THE BOUNDARY DERIVATIVES FOR THE
C                           SPLINE ROUTINE 'XXSPLE', SEE 'XXSPLE'.
C                           (VALID VALUES = <0, 0, 1, 2, 3, 4)
C
C          (L*4)  LSETX   = .TRUE.  => SET UP SPLINE PARAMETERS RELATING
C                                      TO 'XIN' AXIS.
C                           .FALSE. => DO NOT SET UP SPLINE PARAMETERS
C                                      RELATING TO 'XIN' AXIS.
C                                      (I.E. THEY WERE SET IN A PREVIOUS
C                                            CALL )
C                           (VALUE SET TO .FALSE. BY 'XXSPLE')
C
C          (R*8)  R8FUN1  = FUNCTION - (SEE ROUTINES SECTION BELOW)
C          (R*8)  LOGSFR  = LOG( SCALING FACTOR FOR DATA FILE RECEIVER
C                                TEMPERATURES )
C          (R*8)  LOGSFD  = LOG( SCALING FACTOR FOR DATA FILE DONOR
C                                TEMPERATURES )
C          (R*8)  RMR     = RECIPROCAL OF 'RMASS' - USED IN EXTRAPOLATN.
C          (R*8)  DMD     = RECIPROCAL OF 'DMASS' - USED IN EXTRAPOLATN.
C          (R*8)  AMR     = RECIPROCAL OF 'AMSRA' - USED IN EXTRAPOLATN.
C          (R*8)  AMD     = RECIPROCAL OF 'AMSDA' - USED IN EXTRAPOLATN.
C          (R*8)  RDMIN   = FACTOR USED IN  EXTRAPOLATION  OF  DONOR
C                           TEMPERATURES BELOW THE MINIMUM DATA VALUE.
C          (R*8)  RDMAX   = FACTOR USED IN  EXTRAPOLATION  OF  DONOR
C                           TEMPERATURES ABOVE THE MAXIMUM DATA VALUE.
C          (R*8)  RRMIN   = FACTOR USED IN  EXTRAPOLATION  OF  RECEIVER
```

```
C                           TEMPERATURES BELOW THE MINIMUM DATA VALUE.
C          (R*8)  RRMAX   = FACTOR USED IN  EXTRAPOLATION  OF  RECEIVER
C                           TEMPERATURES ABOVE THE MAXIMUM DATA VALUE.
C          (R*8)  VAL1    = VALUE USED IN EXTRAPOLATION.
C          (R*8)  VAL2    = VALUE USED IN EXTRAPOLATION.
C          (R*8)  COEF1   = COEFFICIENT USED TO CALC. EXTRAPOLTED VALUE
C          (R*8)  COEF2   = COEFFICIENT USED TO CALC. EXTRAPOLTED VALUE
C
C          (R*8)  XIN()   = 1) LOG( DATA FILE DONOR    TEMPERATURES )
C                           2) LOG( DATA FILE RECEIVER TEMPERATURES )
C          (R*8)  YIN()   = LOG( DATA FILE RATE-COEFFICIENTS )
C          (R*8)  XOUT()  = 1) LOG( SCALED USER ENTERED DONOR    TEMPS.)
C                           2) LOG( SCALED USER ENTERED RECEIVER TEMPS.)
C          (R*8)  YOUT()  = LOG( OUTPUT GENERATED RATE COEFFICIENTS )
C          (R*8)  YPASS(,)= LOG( RATE COEFFICIENTS ) INTERMEDIATE ARRAY
C                           WHICH STORES INTERPOLATED/EXTRAPOLATED RATE
C                           COEFFICIENT VALUES BEWTEEN THE  TWO  SPLINE
C                           SECTIONS.
C          (R*8)  DF()    = SPLINE INTERPOLATED DERIVATIVES
C
C
C NOTE:
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          XXSPLE     ADAS      SPLINE SUBROUTINE (EXTENDED DIAGNOSTICS)
C          R8FUN1     ADAS      REAL*8 FUNCTION: ( X -> X )
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 4569
C
C DATE:    21/02/91
C
C UNIX-IDL PORT:
C
C AUTHOR:  WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C
C DATE:    22TH MARCH 1996
C
C VERSION: 1.1 DATE: 22-03-96
C MODIFIED: WILLIAM OSBORN
C     - FIRST VERSION
C
C VERSION: 1.2 DATE: 02-03-96
C MODIFIED: WILLIAM OSBORN
C     PROPER HEADER INFORMATION ADDED
C
C VERSION: 1.3 DATE: 7-11-97
C MODIFIED: Martin O'Mullane
C     increased NOUT to 24
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             ITDA,        ITRA,        ITVAL,       NTDDIM
      INTEGER             NTRDIM
      LOGICAL             LTDRNG(ITVAL),            LTRRNG(ITVAL)
      REAL*8              AMSDA,       AMSRA,       DMASS
      REAL*8              QFTCXA(NTDDIM,NTRDIM),    QTCXA(ITVAL)
```

```
REAL*8              RMASS,       TDEVA(ITVAL)
REAL*8              TFDA(ITDA),  TFRA(ITRA),  TREVA(ITVAL)
```

6.22 e5titl: Subroutine e5titl from library adas5xx

```
      SUBROUTINE E5TITL( IBSEL  , DSFULL ,
     &                   CDONOR , CRECVR , CFSTAT,
     &                   TITLX
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: E5TITL *********************
C
C  PURPOSE:  TO CREATE THE DESCRIPTIVE TITLE FOR SELECTED DATA-BLOCK.
C
C  CALLING PROGRAM: ADAS505/SQTCX
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  IBSEL    = SELECTED DATA-BLOCK: INDEX
C  INPUT : (C*80) DSFULL   = FULL INPUT DATA SET NAME
C
C
C  INPUT : (C*9)  CDONOR   = SELECTED DATA-BLOCK: DONOR IDENTITY
C
C  INPUT : (C*9)  CRECVR   = SELECTED DATA-BLOCK: RECEIVER IDENTITY
C
C  INPUT : (C*10) CFSTAT  = SELECTED DATA-BLOCK: FINAL STATE SPEC.
C
C  OUTPUT: (C*120) TITLX    = SELECTED DATA-BLOCK: DESCRIPTIVE TITLE
C
C          (C*2)  C2       = GENERAL USE 2 BYTE  CHARACTER  STRING
C
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/37
C          JET EXT. 2520
C
C DATE:    01/05/91
C
C UNIX-IDL PORT:
C
C AUTHOR:  WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C
C DATE:    21TH MARCH 1996
C
C VERSION: 1.1 DATE: 21-03-96
C MODIFIED: WILLIAM OSBORN
C     - FIRST VERSION. CONVERTED TO NEW DATA FORMAT
C
C VERSION: 1.2 DATE: 02-03-96
C MODIFIED: WILLIAM OSBORN
C     PROPER HEADER INFORMATION ADDED
C
C VERSION: 1.3 DATE: 10-03-96
C MODIFIED: WILLIAM OSBORN
C     REDUNDANT VARIABLES REMOVED
C
C VERSION: 1.4 DATE: 10-03-96
C MODIFIED: WILLIAM OSBORN
C     REDUNDANT COMMENTS REMOVED
C
C VERSION: 1.5 DATE: 22-04-96
```

```
C MODIFIED: TIM HAMMOND
C           - INCREASED DSFULL TO *80
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      CHARACTER*9         CDONOR
      CHARACTER*10        CFSTAT
      CHARACTER*9         CRECVR
      CHARACTER*80        DSFULL
      CHARACTER*120       TITLX
      INTEGER             IBSEL
```

6.23 e6coll: Subroutine e6coll from library adas5xx

```
      SUBROUTINE E6COLL (IUNT20, FILC, DSFULL, ELEM, IZ,
     &                   ISTRN, MAXT, APWLO , SWLO  ,
     &                   TOA, GFTOA)
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: E6COLL *********************
C
C  PURPOSE:  TO STORE SELECTED GFT INTO A COLLECTION FILE.
C
C  CALLING PROGRAM: ADAS506
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  IUNIT20  =  UNIT NUMBER FOR COLLECTION FILE
C  INPUT : (C*80) FILC     =  COLLECTION FILENAME
C  INPUT : (C*80) DSFULL   =  INPUT DATA SET NAME
C  INPUT : (C*2)  ELEM     =  ELEMENT SYMBOL.
C  INPUT : (I*4)  IZ       =  RECOMBINED ION CHARGE
C  INPUT : (I*4)  ISTRN    =  SELECTED TRANSITION INDEX
C  INPUT : (I*4)  MAXT     =  NUMBER OF ISPF ENTERED TEMPERATURE VALUES.
C  INPUT : (R*8)  APWLO    =  SELECTED APPROXIMATE WAVELENGTHS (A)
C  INPUT : (R*8)  SWLO     =  SELECTED EXACT WAVELENGTHS (A)
C  INPUT : (R*8)  TOA()    =  ISPF ENTERED TEMPERATURES (kelvin)
C  INPUT : (R*8)  GFTOA()  =  SPLINE INTEROPLATED GFT VALUE AT 'TOA()'
C
C          (I*4)  NTRAN    =  NUMBER OF G(T) FUNCTION IN FINAL COLL. FILE
C          (R*8)  TX(,)    =  STORED TEMPERATURE FROM OLD COLLECTION FILE
C          (R*8)  GFTX(,)  =  STORED G(T) FROM OLD COLLECTION FILE
C          (C*2)  ELEMC()  =  ELEMENT SYMBOL READ FROM COLL FILE
C          (I*4)  IZC()    =  RECOMBINED ION CHARGE READ FROM COLL FILE
C          (I*4)  ISTRNC() =  TRANSITION INDEX FROM COLL FILE
C          (I*4)  NG()     =  NUMBER OF DATA POINTS FOR G(T) IN COLL FILE
C          (L*4)  LFEXST   = .TRUE. COLLECTION FILE EXISTS
C
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4UNIT     ADAS      FETCH UNIT NUMBER FOR OUTPUT OF MESSAGES
C          R8FCTN     ADAS      CONVERTS FROM CHARACTER TO REAL VARIABLE
C          I4FCTN     ADAS      CONVERTS CHARACTER STRING TO INTEGER
C          I4EIZ0     ADAS      RETURNS Z0 FOR GIVEN ELEMENT SYMBOL
C
C AUTHOR  : Alessandro Lanzafame, University of Strathclyde
C
C DATE:    june06-95
C
C UPDATE:
C VERSION: 1.2 DATE: 09-11-95
C MODIFIED: Alessandro Lanzafame
C           - MTRAN: 100 -> 300
C             MTEMP: 100 -> 101
C
C VERSION       1.3             DATE:   31-05-96
C MODIFIED: Alessandro Lanzafame
C           - MTRAN: 300 -> 800
C
C VERSION       1.4             DATE:   13-10-99
C MODIFIED: Martin O'Mullane
```

```
C           - Replace format statements 1004 and 1005. Not all
C             compilers can use dynamically created format statements.
C
C VERSION       1.5             DATE:   08-07-02
C MODIFIED: Richard Martin
C           - Fixed format statements introduced in 1.4 to work on SUN's
C
C-----------------------------------------------------------------------
      CHARACTER*80        DSFULL
      CHARACTER*2         ELEM
      CHARACTER*80        FILC
      INTEGER             ISTRN,       IUNT20,      IZ,          MAXT
      REAL*8              APWLO,       GFTOA(*),    SWLO,        TOA(*)
```

6.24 e6data: Subroutine e6data from library adas5xx

```
       SUBROUTINE E6DATA( IUNIT  , NDLEV  , NDTEM , NDTRN ,
     &                    ELEM   , IZ     , IZ0   , IZ1   ,
     &                    IL     ,
     &                    IA     , CSTRGA , ISA   , ILA   , XJA    ,
     &                    NV     ,
     &                    TEA    , DENSA  , PRESA , RNHNE , TMA    ,
     &                    ITRAN  ,
     &                    I1A    , I2A    , APWL  , SWL   , GFT    ,
     &                    LVALID , INDX
     &                  )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: E6DATA *********************
C
C  PURPOSE:  TO FETCH DATA FROM INPUT GFT DATA SET OF TYPE ADF20.
C
C  CALLING PROGRAM: ADAS506
C
C           THE UNITS USED IN THE DATA FILE ARE TAKEN AS FOLLOWS:
C
C           ELECTRON TEMPERATURE: KELVIN
C           ELECTRON DENSITY    : CM-3
C           ELECTRON PRESSURE   : K CM-3
C           TIME                : NOT SPECIFIED
C           WAVELENGTH          : ANGSTROM
C           GFT COEFFT.         : CM3 SEC-1
C
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  IUNIT   = UNIT TO WHICH INPUT FILE IS ALLOCATED
C  INPUT : (I*4)  NDLEV   = MAXIMUM NUMBER OF LEVELS THAT CAN BE READ
C  INPUT : (I*4)  NDTEM   = MAXIMUM NUMBER OF TEMPERATURES
C  INPUT : (I*4)  NDTRN   = MAX. NUMBER OF TRANSITIONS THAT CAN BE READ
C
C  OUTPUT: (C*2)  ELEM    = ELEMENT SYMBOL.
C  OUTPUT: (I*4)  IZ      =  RECOMBINED ION CHARGE READ
C  OUTPUT: (I*4)  IZ0     =         NUCLEAR CHARGE READ
C  OUTPUT: (I*4)  IZ1     = RECOMBINING ION CHARGE READ
C                           (NOTE: IZ1 SHOULD EQUAL IZ+1)
C
C  OUTPUT: (I*4)  IL      = INPUT DATA FILE: NUMBER OF ENERGY LEVELS
C
C  OUTPUT: (I*4)  IA()    = ENERGY LEVEL INDEX NUMBER
C  OUTPUT: (C*18) CSTRGA()= NOMENCLATURE/CONFIGURATION FOR LEVEL 'IA()'
C  OUTPUT: (I*4)  ISA()   = MULTIPLICITY FOR LEVEL 'IA()'
C                           NOTE: (ISA-1)/2 = QUANTUM NUMBER (S)
C  OUTPUT: (I*4)  ILA()   = QUANTUM NUMBER (L) FOR LEVEL 'IA()'
C  OUTPUT: (R*8)  XJA()   = QUANTUM NUMBER (J-VALUE) FOR LEVEL 'IA()'
C                           NOTE: (2*XJA)+1 = STATISTICAL WEIGHT
C
C  OUTPUT: (I*4)  NV      = INPUT DATA FILE: NUMBER OF TEMP/DENS/PRESS/
C                           TIME SETS
C  OUTPUT: (R*8)  TEA()   = INPUT DATA FILE: ELECTRON TEMPERATURES (K)
C  OUTPUT: (R*8)  DENSA() = INPUT DATA FILE: ELECTRON DENSITIES (CM-3)
C  OUTPUT: (R*8)  PRESA() = INPUT DATA FILE: ELECTRON PRESSURES (K CM-3)
C  OUTPUT: (R*8)  TMA()   = INPUT DATA FILE: TIMES OR ARBITRARY (S ?)
C
C  OUTPUT: (I*4)  ITRAN   = INPUT DATA FILE: NUMBER OF TRANSITIONS
```

```
C
C  OUTPUT: (I*4)  I1A()   = TRANSITION:
C                           LOWER ENERGY LEVEL INDEX
C  OUTPUT: (I*4)  I2A()   = TRANSITION:
C                           UPPER ENERGY LEVEL INDEX
C  OUTPUT: (R*8)  APWL()  = APPROXIMATE TRANSITION WAVELENGTH (A)
C  OUTPUT: (R*8)  SWL()   = EXACT TRANSITION WAVELENGTH (A)  (IF SET)
C  OUTPUT: (R*8)  GFT(,)  = GFT COEFFICIENT FOR TRANSITION (CM3 S-1)
C                           1ST DIMENSION - TEMPERATURE 'TEA()'
C                           2ND DIMENSION - TRANSITION INDEX
C  OUTPUT: (L*4)  LVALID  = .TRUE. DATA SET READ AND APPEARS VALID
C                         = .FALSE. ERROR DETECTED IN READING DATA SET
C#
C# OUTPUT: (I*4)  INDX()  = TRANSITION INDEX (USED IN DEM CODES)
C
C
C          (I*4)  I4UNIT  = FUNCTION (SEE ROUTINE SECTION BELOW)
C          (I*4)  I4FCTN  = FUNCTION (SEE ROUTINE SECTION BELOW)
C          (I*4)  I4EIZ0  = FUNCTION (SEE ROUTINE SECTION BELOW)
C          (I*4)  I       = GENERAL USE.
C          (I*4)  IABT    = RETURN CODE FROM 'I4FCTN'
C          (I*4)  J       = GENERAL USE.
C          (I*4)  K       = GENERAL USE.
C          (I*4)  NVAL    = GENERAL USE
C          (I*4)  IRECL   = RECORD LENGTH OF INPUT DATASET (<=128)
C
C
C          (C*1)  CSLASH  = '/' - DELIMITER FOR 'XXHKEY'
C          (C*4)  C4      = GENERAL USE FOUR BYTE CHARACTER
C          (C*5)  IONNAM  = EMITTING ION READ FROM DATA SET
C          (C*7)  CKEY1   = 'NLEVELS' - INPUT HEADER KEY
C          (C*6)  CKEY2   = 'NKNOTS'  - INPUT HEADER KEY
C          (C*6)  CKEY3   = 'NLINES'  - INPUT HEADER KEY
C          (C*3)  TITLED  = ELEMENT SYMBOL INCLUDING '+'
C          (C*80) CLINE   = CURRENT ENERGY LEVEL INDEX PARAMETER LINE
C          (C*127)BUFFER  = GENERAL STRING BUFFER STORAGE
C
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          XXWORD     ADAS      EXTRACT POSITION OF NUMBER IN BUFFER
C          I4UNIT     ADAS      FETCH UNIT NUMBER FOR OUTPUT OF MESSAGES
C          R8FCTN     ADAS      CONVERTS FROM CHARACTER TO REAL VARIABLE
C          I4FCTN     ADAS      CONVERTS CHARACTER STRING TO INTEGER
C          I4EIZ0     ADAS      RETURNS Z0 FOR GIVEN ELEMENT SYMBOL
C
C
C ROUTINES: NONE
C
C AUTHOR:  H. P. SUMMERS, JET
C          K1/1/57
C          JET EXT. 4941
C
C DATE:    07/04/94
C
C UPDATE:  APR18-95
C#         A. C. LANZAFAME, DPAP UNIVERSITY OF STRATHCLYDE
C#         TRANSITION INDEX (INDX) ADDED. USED IN DEM CODES
C#         TO IDENTIFY THE TRANSITION
C#
```

```
C#         CHARACTER CSTRGA(NDLEV)*(*) changed to CHARACTER*18 CSTRGA(NDLEV)
C#         after experinced unstable behaviour on Sun workstation
C
C UPDATE:
C VERSION:      1.2             DATE:   09-11-95
C MODIFIED: Alessandro Lanzafame
C           - Commented out superfluous variables
C
C
C VERSION: 1.3                          DATE: 06-06-2003
C MODIFIED: Martin O'Mullane
C                   - Warn user that the routine is now deprecated
C                     and that xxdata_20 should be used instead.
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      CHARACTER*18        CSTRGA(NDLEV)
      CHARACTER*2         ELEM
      INTEGER             I1A(NDTRN),  I2A(NDTRN),  IA(NDLEV),   IL
      INTEGER             ILA(NDLEV),  INDX(NDTRN), ISA(NDLEV),  ITRAN
      INTEGER             IUNIT,       IZ,          IZ0,         IZ1
      INTEGER             NDLEV,       NDTEM,       NDTRN,       NV
      LOGICAL             LVALID
      REAL*8              APWL(NDTRN), DENSA(NDTEM)
      REAL*8              GFT(NDTEM,NDTRN),         PRESA(NDTEM)
      REAL*8              RNHNE(NDTEM),             SWL(NDTRN)
      REAL*8              TEA(NDTEM),  TMA(NDTEM),  XJA(NDLEV)
```

6.25 e6spln: Subroutine e6spln from library adas5xx

```
      SUBROUTINE E6SPLN(         LOSEL ,
     &                    NV    , MAXT  , NPSPL  ,
     &                    SCEF  , TOA   , TOSA   ,
     &                    GOFTA , GFTOA , GFTOSA ,
     &                    LTRNG
     &                  )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: E6SPLN *********************
C
C  (IDENTICAL TO: C1SPLN (EXCEPT SOME VARIABLE NAMES ARE CHANGED))
C
C  PURPOSE:
C         1) PERFORMS CUBIC SPLINE ON LOG(TEMPERATURE) VERSUS LOG(GFT)
C            INPUT DATA. ('SCEF' VERSUS 'GOFTA' , NV DATA PAIRS)
C
C         2) INTERPOLATES 'MAXT'  GFT VALUES USING ABOVE SPLINES AT
C            TEMPERATURES READ IN FROM ISPF PANELS FOR TABULAR OUTPUT.
C            (ANY TEMPERATURE VALUES WHICH REQUIRED EXTRAPOLATION TO
C             TAKE PLACE ARE SET TO ZERO).
C                 - THIS STEP ONLY TAKES PLACE IF 'LOSEL=.TRUE.' -
C
C         3) INTERPOLATES 'NPSPL' GFT VALUES USING ABOVE SPLINES AT
C            TEMPERATURES EQUI-DISTANCE ON RANGE OF LOG(TEMPERATURES)
C            STORED IN INPUT 'SCEF' ARRAY.
C
C  CALLING PROGRAM: ADAS506
C
C
C  SUBROUTINE:
C
C  INPUT : (L*4)  LOSEL   = .TRUE.  => CALCULATE GFTS FOR INPUT TEMPS.
C                                      READ FROM ISPF PANEL.
C                           .FALSE. => - DO NOT DO THE ABOVE -
C
C  INPUT : (I*4)  NV      = INPUT DATA FILE: NUMBER OF GFT/TEMPERATURE
C                           PAIRS READ FOR THE TRANSITION BEING ASSESSED
C  INPUT : (I*4)  MAXT    = NUMBER OF ISPF ENTERED TEMPERATURE VALUES AT
C                           WHICH INTERPOLATED GFT VALUES ARE REQUIRED
C                           FOR TABULAR OUTPUT.
C  INPUT : (I*4)  NPSPL   = NUMBER OF  SPLINE  INTERPOLATED  GFT/TEMP.
C                           REQUIRED FOR GRAPHICAL DISPLAY.
C
C  INPUT : (I*4)  SCEF()  = INPUT DATA FILE: TEMPERATURES (KELVIN)
C  INPUT : (I*4)  TOA()   = ISPF PANEL ENTERED TEMPERATURES (KELVIN)
C  OUTPUT: (I*4)  TOSA()  = 'NPSPL' TEMPERATURES FOR GRAPHICAL OUTPUT
C                           (KELVIN).
C
C  INPUT : (R*8)  GOFTA() = INPUT DATA FILE: SELECTED TRANSITION -
C                           GFT VALUES AT 'SCEF()'.
C  OUTPUT: (I*4)  GFTOA() = SPLINE INTERPOLATED GFT VALUES AT 'TOA()'
C                           (EXTRAPOLATED VALUES = 0.0).
C  OUTPUT: (R*8)  GFTOSA()= SPLINE INTERPOLATED GFT VALUES AT 'TOSA()'
C
C  OUTPUT: (L*4)  LTRNG() = .TRUE.  => OUTPUT   SPLINE    VALUE    WAS
C                                      INTERPOLATED FOR 'DLOG(TOA()'.
C                           .FALSE. => OUTPUT   SPLINE    VALUE    WAS
C                                      EXTRAPOLATED FOR 'DLOG(TOA()'.
C                                      (NOTE: 'YOUT()=0' AS 'IOPT < 0').
```

```
C
C          (I*4)  NIN     = PARAMETER = MAX. NO. OF  INPUT  TEMP/GFT
C                                       PAIRS MUST BE >= 'NV'
C          (I*4)  NOUT    = PARAMETER = MAX. NO. OF 'OUTPUT TEMP/GFT
C                                       PAIRS MUST BE >= 'MAXT' & 'NPSPL'
C
C          (I*4)  IARR    = ARRAY SUBSCRIPT USED FOR TEMP/GFT PAIRS
C          (I*4)  IOPT    = DEFINES THE BOUNDARY DERIVATIVES FOR THE
C                           SPLINE ROUTINE 'XXSPLE', SEE 'XXSPLE'.
C                           (VALID VALUES = <0, 0, 1, 2, 3, 4)
C
C          (R*8)  TSTEP   = THE SIZE OF STEP BETWEEN 'XOUT()' VALUES FOR
C                           GRAPHICAL  OUTPUT  TEMP/GFT  PAIRS  TO  BE
C                           CALCULATED USING SPLINES.
C
C          (L*4)  LSETX   = .TRUE.  => SET UP SPLINE PARAMETERS RELATING
C                                      TO 'XIN' AXIS.
C                           .FALSE. => DO NOT SET UP SPLINE PARAMETERS
C                                      RELATING TO 'XIN' AXIS.
C                                      (I.E. THEY WERE SET IN A PREVIOUS
C                                            CALL )
C                           (VALUE SET TO .FALSE. BY 'XXSPLE')
C
C          (R*8)  XIN()   = LOG( 'SCEF()' )
C          (R*8)  YIN()   = LOG( 'GOFTA()' )
C          (R*8)  XOUT()  = LOG(TEMPERATURES AT WHICH SPLINES REQUIRED)
C          (R*8)  YOUT()  = LOG(OUTPUT SPLINE INTERPOLATED GFT VALUES)
C          (R*8)  DF()    = SPLINE INTERPOLATED DERIVATIVES
C
C          (L*4)  LDUMP() = .TRUE.  => OUTPUT SPLINE VALUE INTRPOLATED
C                                      FOR 'YOUT()'.
C                           .FALSE. => OUTPUT SPLINE VALUE EXTRAPOLATED
C                                      FOR 'YOUT()'.
C                                      (NOTE: USED AS A DUMMY ARGUMENT.
C                                             ALL VALUES WILL BE TRUE.)
C
C NOTE:
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          XXSPLE     ADAS      SPLINE SUBROUTINE (EXTENDED DIAGNOSTICS)
C          R8FUN1     ADAS      REAL*8 FUNCTION: ( X -> X )
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 4569
C
C DATE:    05/02/91
C
C VERSION: 1.2 DATE: 09-11-95
C MODIFIED: ALESSANDRO LANZAFAME
C               - NIN: 100 -> 101
C                 NOUT: 100 -> 101
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             MAXT,        NPSPL,       NV
      LOGICAL             LOSEL,       LTRNG(MAXT)
      REAL*8              GFTOA(MAXT), GFTOSA(NPSPL)
```

```
REAL*8              GOFTA(NV),    SCEF(NV),     TOA(MAXT)
REAL*8              TOSA(NPSPL)
```

6.26 e6tran: Subroutine e6tran from library adas5xx

```
       SUBROUTINE E6TRAN( NDLEV  , NDTRN  , NDTEM ,
     &                    IL     , ISTRN  , NV    ,
     &                    IA     , XJA    ,
     &                    APWL   , SWL    ,
     &                    I1A    , I2A    ,
     &                    GFTA   ,
     &                    IUPPER , ILOWER ,
     &                    LUPPER , LLOWER ,
     &                    WUPPER , WLOWER ,
     &                    APWLO  , SWLO   ,
     &                    GOFTA
     &                  )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: E6TRAN *********************
C
C  PURPOSE:  TO SET UP SELECTED TRANSITION PARAMETERS.
C
C  CALLING PROGRAM: ADAS506
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  NDLEV   = MAXIMUM NUMBER OF INDEX LEVELS
C  INPUT : (I*4)  NDTRN   = MAXIMUM NUMBER OF TRANSITIONS
C  INPUT : (I*4)  NDTEM   = MAXIMUM NUMBER OF INPUT FILE TEMPERATURES
C
C  INPUT : (I*4)  IL      = NUMBER OF INDEX LEVELS
C  INPUT : (I*4)  ISTRN   = SELECTED TRANSITION INDEX.
C  INPUT : (I*4)  NV      = INPUT DATA FILE: NUMBER OF GFT/TEMPERATURE
C                           PAIRS FOR THE SELECTED TRANSITION.
C
C  INPUT : (I*4)  IA()    = LEVEL INDEX NUMBER ARRAY
C  INPUT : (R*8)  XJA()   = QUANTUM NUMBER (J-VALUE) FOR LEVEL
C                           NOTE: (2*XJA)+1 = STATISTICAL WEIGHT
C
C  INPUT : (R*8)  APWL()  = APPROXIMATE WAVELENGTHS OF TRANSITIONS (A)
C  INPUT : (R*8)  SWL()   = EXACT WAVELENGTHS OF TRANSITIONS (A)
C  INPUT : (I*4)  I1A()   = LOWER LEVEL INDEX FOR RADIATIVE
C                           TRANSITION
C  INPUT : (I*4)  I2A()   = UPPER LEVEL INDEX FOR RADAITIVE
C                           TRANSITION
C  INPUT : (I*4)  GFTA(,) = GFT VALUES FOR RADIATIVE TRANSITION
C                            1st DIMENSION: TEMPERATURE INDEX
C                            2nd DIMENSION: TRANSITION INDEX
C
C  OUTPUT: (I*4)  IUPPER  = SELECTED TRANSITION: UPPER LEVEL ARRAY INDEX
C  OUTPUT: (I*4)  ILOWER  = SELECTED TRANSITION: LOWER LEVEL ARRAY INDEX
C
C
C  OUTPUT: (I*4)  LUPPER  = SELECTED TRANSITION: UPPER INDEX LEVEL
C  OUTPUT: (I*4)  LLOWER  = SELECTED TRANSITION: LOWER INDEX LEVEL
C
C  OUTPUT: (R*8)  WUPPER  = SELECTED TRANSITION: UPPER LEVEL STAT. WT.
C  OUTPUT: (R*8)  WLOWER  = SELECTED TRANSITION: LOWER LEVEL STAR. WT.
C                           (NOTE: STAT. WT. = STATISTICAL WEIGHT)
C  OUTPUT: (R*8)  APWLO   = SELECTED APPROXIMATE WAVELENGTH (A)
C  OUTPUT: (R*8)  SWLO    = SELECTED EXACT WAVELENGTH (A)
C
C  OUTPUT: (R*8)  GOFTA() = INPUT DATA FILE: SELECTED TRANSITION -
```

```
C                                              GFT VALUE AT 'TEMP()'
C
C          (I*4)  I       = GENERAL USE.
C
C ROUTINES: NONE
C
C AUTHOR:  H. P .SUMMERS, JET
C          K1/1/57
C          JET EXT. 4941
C
C DATE:    08/04/94
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      INTEGER             I1A(NDTRN),  I2A(NDTRN),  IA(NDLEV),    IL
      INTEGER             ILOWER,      ISTRN,       IUPPER,       LLOWER
      INTEGER             LUPPER,      NDLEV,       NDTEM,        NDTRN
      INTEGER             NV
      REAL*8              APWL(NDTRN), APWLO,       GFTA(NDTEM,NDTRN)
      REAL*8              GOFTA(NDTEM),             SWL(NDTRN),   SWLO
      REAL*8              WLOWER,      WUPPER,      XJA(NDLEV)
```

6.27 e7data: Subroutine e7data from library adas5xx

```
       SUBROUTINE E7DATA( IUNIT  , DSNAME ,
     &                    NSTORE , NTDIM  ,
     &                    ESYM   , IZ0    ,
     &                    NBSEL  , ISELA  ,
     &                    IZ     , IZ1    ,
     &                    CWAVEL , CIION  , CICODE , CISCRP , CITYPE ,
     &                    ITA    ,
     &                    TMA    , TETA   , DENSA  , GCF
     &                  )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: E7DATA *********************
C
C  PURPOSE:  TO  FETCH  DATA  FROM  INPUT CONTRIBUTION FUNCTIONS
C             OF AN ELEMENT AND ITS IONS.
C            (MEMBER STORED IN IONELEC.DATA - MEMBER PREFIX 'GCF#').
C
C  CALLING PROGRAM: ADAS507/SGCF
C
C  DATA:
C
C           UP TO 'NSTORE' SETS (DATA-BLOCKS) OF DATA MAY BE  READ FROM
C           THE FILE - EACH BLOCK FORMING A COMPLETE SET OF CONTRIBUTION
C           FUNCTION VALUES FOR GIVEN TEMPERATURES.
C           EACH DATA-BLOCK  IS  ANALYSED INDEPENDENTLY OF ANY  OTHER
C           DATA-BLOCK.
C
C           THE UNITS USED IN THE DATA FILE ARE TAKEN AS FOLLOWS:
C
C           MODEL PARAMETER     : UNSPECIFIED
C           DENSITIES           : CM-3
C           TEMPERATURES        : EV
C           CONTR. FUNCTION     : CM**3 S-1
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  IUNIT    = UNIT TO WHICH INPUT FILE IS ALLOCATED.
C  INPUT : (C*44) DSNAME   = MVS DATA SET NAME OF DATA SET BEING READ
C
C  INPUT : (I*4)  NSTORE   = MAXIMUM NUMBER  OF  INPUT DATA-BLOCKS  THAT
C                            CAN BE STORED.
C  INPUT : (I*4)  NTDIM    = MAX NUMBER OF ELECTRON TEMPERATURES ALLOWED
C
C  OUTPUT: (C*2)  ESYM     = READ - RADIATING ION - ELEMENT SYMBOL
C  OUTPUT: (I*4)  IZ0      = READ - RADIATING ION - NUCLEAR CHARGE
C
C  OUTPUT: (I*4)  NBSEL    = NUMBER OF DATA-BLOCKS ACCEPTED & READ IN.
C  OUTPUT: (I*4)  ISELA()  = READ - DATA-SET DATA-BLOCK ENTRY INDICES
C                            DIMENSION: DATA-BLOCK INDEX
C
C  OUTPUT: (I*4)  IZ()     = READ - RADIATING ION CHARGE
C                                   ( SET TO -1 IF WHOLE ELEMENT)
C                            DIMENSION: DATA-BLOCK INDEX
C  OUTPUT: (I*4)  IZ1()    = READ - RADIATING ION CHARGE +1
C                                   ( SET TO 1 IF WHOLE ELEMENT)
C                            DIMENSION: DATA-BLOCK INDEX
C
C  OUTPUT: (C*10) CWAVEL() = READ - WAVELENGTH (ANGSTROMS)
C                            DIMENSION: DATA-BLOCK INDEX
```

```
C  OUTPUT: (C*5)  CIION()  = READ - RADIATING ION (AS <ESYM>+(IZ()> )
C                           DIMENSION: DATA-BLOCK INDEX
C  OUTPUT: (C*8)  CICODE() = READ - SOURCE PROGRAM
C                           DIMENSION: DATA-BLOCK INDEX
C  OUTPUT: (C*8)  CISCRP() = READ - SOURCE SCRIPT
C                           DIMENSION: DATA-BLOCK INDEX
C  OUTPUT: (C*5)  CITYPE() = READ - RADIATION TYPE
C                           DIMENSION: DATA-BLOCK INDEX
C
C  OUTPUT: (I*4)  ITA()    = READ - NUMBER OF ELECTRON TEMPERATURES
C                           DIMENSION: DATA-BLOCK INDEX
C
C  OUTPUT: (R*8)  TMA(,)   = READ - MODEL PARAMETER (UNITS: UNDEFINED)
C                           1st DIMENSION: ELECTRON TEMPERATURE INDEX
C                           2nd DIMENSION: DATA-BLOCK INDEX
C
C  OUTPUT: (R*8)  TETA(,)  = READ - ELECTRON TEMPERATURES (UNITS: EV)
C                           1st DIMENSION: ELECTRON TEMPERATURE INDEX
C                           2nd DIMENSION: DATA-BLOCK INDEX
C
C  OUTPUT: (R*8)  DENSA(,) = READ - ELECTRON DENSITIES    (UNITS: CM-3)
C                           1st DIMENSION: ELECTRON TEMPERATURE INDEX
C                           2nd DIMENSION: DATA-BLOCK INDEX
C
C  OUTPUT: (R*8)  GCF(,)    =READ - FULL SET OF GENERALISED CONTIBUTION
C                                   FUNCTIONS (CM**3 S-1)
C                           1st DIMENSION: ELECTRON TEMPERATURE INDEX
C                           2nd DIMENSION: DATA-BLOCK INDEX
C
C          (I*4)  I4EIZ0   = FUNCTION - (SEE ROUTINES SECTION BELOW)
C          (I*4)  I4FCTN   = FUNCTION - (SEE ROUTINES SECTION BELOW)
C          (I*4)  I4UNIT   = FUNCTION - (SEE ROUTINES SECTION BELOW)
C          (I*4)  IBLK     = ARRAY INDEX: DATA-BLOCK INDEX
C          (I*4)  ITT      = ARRAY INDEX: ELECTRON TEMPERATURE INDEX
C          (I*4)  NTNUM    = NUMBER OF ELECTRON TEMPERATURES FOR CURRENT
C                           DATA-BLOCK
C          (I*4)  IABT     = RETURN CODE FROM 'I4FCTN'
C          (I*4)  IPOS1    = GENERAL USE STRING INDEX VARIABLE
C          (I*4)  IPOS2    = GENERAL USE STRING INDEX VARIABLE
C
C          (R*8)  R8FCTN   = FUNCTION - (SEE ROUTINES SECTION BELOW)
C
C          (L*4)  LBEND    = IDENTIFIES WHETHER THE LAST OF THE  INPUT
C                           DATA SUB-BLOCKS HAS BEEN LOCATED.
C                           (.TRUE. => END OF SUB-BLOCKS REACHED)
C
C          (C*1)  CSLASH   = '/' - DELIMITER FOR 'XXHKEY'
C          (C*2)  C2       = GENERAL USE TWO BYTE CHARACTER STRING
C          (C*4)  CKEY1    = 'CODE'  - INPUT BLOCK HEADER KEY
C          (C*6)  CKEY2    = 'SCRIPT - INPUT BLOCK HEADER KEY
C          (C*4)  CKEY3    = 'TYPE'  - INPUT BLOCK HEADER KEY
C          (C*4)  CKEY4    = 'ISEL'  - INPUT BLOCK HEADER KEY
C          (C*10) C10      = GENERAL USE TEN BYTE CHARACTER STRING
C          (C*80) C80      = GENERAL USE 80 BYTE  CHARACTER  STRING  FOR
C                           THE INPUT OF DATA-SET RECORDS.
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          XXHKEY     ADAS      OBTAIN KEY/RESPONSE STRINGS FROM TEXT
C          I4EIZ0     ADAS      INTEGER*4 FUNCTION    -
```

```
C                               RETURNS Z0 FOR GIVEN ELEMENT SYMBOL
C          I4FCTN     ADAS      INTEGER*4 FUNCTION    -
C                               CONVERT CHARACTER STRING TO INTEGER
C          I4UNIT     ADAS      INTEGER*4 FUNCTION    -
C                               FETCH UNIT NUMBER FOR OUTPUT OF MESSAGES
C          R8FCTN     ADAS      REAL*8 FUNCTION       -
C                               CONVERT CHARACTER STRING TO REAL*8
C
C AUTHOR:  H. P. SUMMERS, JET
C          K1/1/57
C          JET EXT. 4941
C
C DATE:    15/04/94
C
C UPDATE:
C
C-----------------------------------------------------------------------
      CHARACTER*8         CICODE(NSTORE)
      CHARACTER*5         CIION(NSTORE)
      CHARACTER*8         CISCRP(NSTORE)
      CHARACTER*5         CITYPE(NSTORE)
      CHARACTER*10        CWAVEL(NSTORE)
      CHARACTER*80        DSNAME
      CHARACTER*2         ESYM
      INTEGER             ISELA(NSTORE),            ITA(NSTORE), IUNIT
      INTEGER             IZ(NSTORE),  IZ0,         IZ1(NSTORE), NBSEL
      INTEGER             NSTORE,      NTDIM
      REAL*8              DENSA(NTDIM,NSTORE),      GCF(NTDIM,NSTORE)
      REAL*8              TETA(NTDIM,NSTORE),       TMA(NTDIM,NSTORE)
```

6.28 e7setp: Subroutine e7setp from library adas5xx

```
       SUBROUTINE E7SETP( NBSEL )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: E7SETP *********************
C
C  PURPOSE:  TO SET UP PARAMETERS IN THE SHARED POOLED FOR PANEL DISPLAY
C            *** IDENTICAL TO: E1SETP
C            *** IDENTICAL TO: C3SETP (EXCEPT 'SNCOMB' -> 'SLINES')
C
C  CALLING PROGRAM: ADAS507
C
C  DATA:
C           DATA IS OBTAINED VIA SUBROUTINE 'E7DATA'
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  NBSEL   = NUMBER OF IONIZING ION COMBINATIONS READ IN.
C                           I.E. NUMBER OF DATA-BLOCKS.
C
C          (C*8)  F6      = PARAMETER = 'VREPLACE'
C
C          (I*4)  ILEN    = LENGTH, IN BYTES, OF ISPF DIALOG VARIABLES
C
C          (C*3)  SLINES  = 'NBSEL'
C          (C*8)  CHA     = FUNCTION POOL DIALOG NAME FOR 'NBSEL'.
C
C ROUTINES:
C           ROUTINE     SOURCE      BRIEF DESCRIPTION
C           ------------------------------------------------------------
C           ISPLNK      ISPF        ISPF PANEL SERVICE ROUTINE
C
C AUTHOR:  H. P. SUMMERS, JET
C          K1/1/57
C          JET EXT. 4941
C
C DATE:    15/04/94
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      INTEGER            NBSEL
```

6.29 e7spln: Subroutine e7spln from library adas5xx

```
      SUBROUTINE E7SPLN(         LOSEL ,
     &                    NV    , MAXT  , NPSPL  ,
     &                    SCEF  , TOA   , TOSA   ,
     &                    GCF   , GCFOA , GCFOSA ,
     &                    LTRNG
     &                  )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: E7SPLN *********************
C
C  (IDENTICAL TO: C1SPLN (EXCEPT SOME VARIABLE NAMES ARE CHANGED))
C
C  PURPOSE:
C         1) PERFORMS CUBIC SPLINE ON LOG(TEMPERATURE) VERSUS LOG(GCF)
C            INPUT DATA. ('SCEF' VERSUS 'GCF' , NV DATA PAIRS)
C
C         2) INTERPOLATES 'MAXT'  GCF VALUES USING ABOVE SPLINES AT
C            TEMPERATURES READ IN FROM ISPF PANELS FOR TABULAR OUTPUT.
C            (ANY TEMPERATURE VALUES WHICH REQUIRED EXTRAPOLATION TO
C             TAKE PLACE ARE SET TO ZERO).
C                 - THIS STEP ONLY TAKES PLACE IF 'LOSEL=.TRUE.' -
C
C         3) INTERPOLATES 'NPSPL' GCF VALUES USING ABOVE SPLINES AT
C            TEMPERATURES EQUI-DISTANCE ON RANGE OF LOG(TEMPERATURES)
C            STORED IN INPUT 'SCEF' ARRAY.
C
C  CALLING PROGRAM: ADAS507
C
C
C  SUBROUTINE:
C
C  INPUT : (L*4)  LOSEL   = .TRUE.  => CALCULATE GCFS FOR INPUT TEMPS.
C                                      READ FROM ISPF PANEL.
C                           .FALSE. => - DO NOT DO THE ABOVE -
C
C  INPUT : (I*4)  NV      = INPUT DATA FILE: NUMBER OF GCF/TEMPERATURE
C                           PAIRS READ FOR THE TRANSITION BEING ASSESSED
C  INPUT : (I*4)  MAXT    = NUMBER OF ISPF ENTERED TEMPERATURE VALUES AT
C                           WHICH INTERPOLATED GCF VALUES ARE REQUIRED
C                           FOR TABULAR OUTPUT.
C  INPUT : (I*4)  NPSPL   = NUMBER OF  SPLINE  INTERPOLATED  GCF/TEMP.
C                           REQUIRED FOR GRAPHICAL DISPLAY.
C
C  INPUT : (I*4)  SCEF()  = INPUT DATA FILE: TEMPERATURES (KELVIN)
C  INPUT : (I*4)  TOA()   = ISPF PANEL ENTERED TEMPERATURES (KELVIN)
C  OUTPUT: (I*4)  TOSA()  = 'NPSPL' TEMPERATURES FOR GRAPHICAL OUTPUT
C                           (KELVIN).
C
C  INPUT : (R*8)  GCF()   = INPUT DATA FILE: SELECTED TRANSITION -
C                           GCF VALUES AT 'SCEF()'.
C  OUTPUT: (I*4)  GCFOA() = SPLINE INTERPOLATED GCF VALUES AT 'TOA()'
C                           (EXTRAPOLATED VALUES = 0.0).
C  OUTPUT: (R*8)  GCFOSA()= SPLINE INTERPOLATED GCF VALUES AT 'TOSA()'
C
C  OUTPUT: (L*4)  LTRNG() = .TRUE.  => OUTPUT   SPLINE    VALUE    WAS
C                                      INTERPOLATED FOR 'DLOG(TOA()'.
C                           .FALSE. => OUTPUT   SPLINE    VALUE    WAS
C                                      EXTRAPOLATED FOR 'DLOG(TOA()'.
C                                      (NOTE: 'IOPT = 0').
```

```
C
C          (I*4)  NIN     = PARAMETER = MAX. NO. OF  INPUT  TEMP/GCF
C                                       PAIRS MUST BE >= 'NV'
C          (I*4)  NOUT    = PARAMETER = MAX. NO. OF 'OUTPUT TEMP/GCF
C                                       PAIRS MUST BE >= 'MAXT' & 'NPSPL'
C          (R*8)  EXPCUT  = PARAMETER = CUT-OFF IN MAGNITUDE OF
C                                       EXPONENT IN FORMEING EXPONENTIAL
C
C          (I*4)  IARR    = ARRAY SUBSCRIPT USED FOR TEMP/GCF PAIRS
C          (I*4)  IOPT    = DEFINES THE BOUNDARY DERIVATIVES FOR THE
C                           SPLINE ROUTINE 'XXSPLE', SEE 'XXSPLE'.
C                           (VALID VALUES = <0, 0, 1, 2, 3, 4)
C
C          (R*8)  TSTEP   = THE SIZE OF STEP BETWEEN 'XOUT()' VALUES FOR
C                           GRAPHICAL  OUTPUT  TEMP/GCF  PAIRS  TO  BE
C                           CALCULATED USING SPLINES.
C
C          (L*4)  LSETX   = .TRUE.  => SET UP SPLINE PARAMETERS RELATING
C                                      TO 'XIN' AXIS.
C                           .FALSE. => DO NOT SET UP SPLINE PARAMETERS
C                                      RELATING TO 'XIN' AXIS.
C                                      (I.E. THEY WERE SET IN A PREVIOUS
C                                            CALL )
C                           (VALUE SET TO .FALSE. BY 'XXSPLE')
C
C          (R*8)  XIN()   = LOG( 'SCEF()' )
C          (R*8)  YIN()   = LOG( 'GCF()' )
C          (R*8)  XOUT()  = LOG(TEMPERATURES AT WHICH SPLINES REQUIRED)
C          (R*8)  YOUT()  = LOG(OUTPUT SPLINE INTERPOLATED GCF VALUES)
C          (R*8)  DF()    = SPLINE INTERPOLATED DERIVATIVES
C
C          (L*4)  LDUMP() = .TRUE.  => OUTPUT SPLINE VALUE INTRPOLATED
C                                      FOR 'YOUT()'.
C                           .FALSE. => OUTPUT SPLINE VALUE EXTRAPOLATED
C                                      FOR 'YOUT()'.
C                                      (NOTE: USED AS A DUMMY ARGUMENT.
C                                             ALL VALUES WILL BE TRUE.)
C
C NOTE:
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          XXSPLE     ADAS      SPLINE SUBROUTINE (EXTENDED DIAGNOSTICS)
C          R8FUN1     ADAS      REAL*8 FUNCTION: ( X -> X )
C
C AUTHOR:  H. P. SUMMERS, JET
C          K1/1/57
C          JET EXT. 4941
C
C DATE:    19/04/94
C
C VERSION  : 1.1
C DATE     : 25-01-2001
C MODIFIED : Martin O'Mullane
C              - Ported to unix from JETSHP.ADASXX50.FORT
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             MAXT,        NPSPL,       NV
```

```
LOGICAL            LOSEL,       LTRNG(MAXT)
REAL*8             GCF(NV),     GCFOA(MAXT), GCFOSA(NPSPL)
REAL*8             SCEF(NV),    TOA(MAXT),   TOSA(NPSPL)
```

6.30 e7titl: Subroutine e7titl from library adas5xx

```
      SUBROUTINE E7TITL( IBSEL  , DSNAME ,
     &                   CWAVEL , CIION  , CICODE , CISCRP , CITYPE ,
     &                   TITLX
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: E7TITL *********************
C
C  PURPOSE:  TO CREATE THE DESCRIPTIVE TITLE FOR SELECTED DATA-BLOCK.
C
C  CALLING PROGRAM: ADAS507/SGCF
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  IBSEL    = SELECTED DATA-BLOCK: INDEX
C  INPUT : (C*44) DSFULL   = FULL MVS INPUT DATA SET NAME
C
C
C  INPUT : (C*10) CWAVEL   = SELECTED DATA-BLOCK: WAVELENGTH
C  INPUT : (C*5)  CIION    = SELECTED DATA-BLOCK: RADIATING ION
C  INPUT : (C*8)  CICODE   = SELECTED DATA-BLOCK: SOURCE CODE
C  INPUT : (C*8)  CISCRP   = SELECTED DATA-BLOCK: SOURCE SCRIPT
C  INPUT : (C*5)  CITYPE   = SELECTED DATA-BLOCK: RADIATION TYPE
C
C  OUTPUT: (C*80) TITLX    = SELECTED DATA-BLOCK: DESCRIPTIVE TITLE
C
C          (C*2)  C2       = GENERAL USE 2 BYTE  CHARACTER  STRING
C
C ROUTINES: NONE
C
C AUTHOR:  H. P. SUMMERS, JET
C          K1/1/57
C          JET EXT. 4941
C
C DATE:    18/04/94
C
C VERSION  : 1.1
C DATE     : 25-01-2001
C MODIFIED : Martin O'Mullane
C              - Ported to unix from JETSHP.ADASXX50.FORT
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      CHARACTER*8         CICODE
      CHARACTER*5         CIION
      CHARACTER*8         CISCRP
      CHARACTER*5         CITYPE
      CHARACTER*10        CWAVEL
      CHARACTER*80        DSNAME
      CHARACTER*120       TITLX
      INTEGER             IBSEL
```

6.31 e9chkb: Subroutine e9chkb from library adas5xx

```
C
      SUBROUTINE E9CHKB( IUNIT  , NBSEL  , IBSEL  ,
     &                   IZ0IN  , IZ0    ,
     &                   LOPEN  , IRCODE
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: E9CHKB *********************
C
C  PURPOSE: TO CHECK THE SELECTED BLOCK (IBSEL) OF DATA  EXISTS  IN  THE
C           INPUT DATA SET
C
C           IF SO IT REPRESENTS THE ENTERED VALUES OF 'IZ0IN'
C           (NUCLEAR CHARGE OF SELECTED IONISING ION ELEMENT).
C
C           IT ALSO CLOSES THE INPUT DATA SET ALLOCATION IF OPEN.
C
C  CALLING PROGRAM: SSCX
C
C  SUBROUTINE:
C
C  INPUT : (I*4)   IUNIT   = UNIT TO WHICH INPUT DATA SET IS ALLOCATED
C  INPUT : (I*4)   NBSEL   = TOTAL NUMBER OF DATA-BLOCKS READ FROM INPUT
C                            DATA SET.
C  INPUT : (I*4)   IBSEL   = INDEX OF DATA-BLOCK SELECTED FOR ANALYSIS
C
C  INPUT : (I*4)   IZ0IN   = REQUESTED: NUCLEAR CHARGE OF ELEMENT
C  INPUT : (I*4)   IZ0     = INPUT FILE: NUCLEAR CHARGE OF ELEMENT
C
C  I/O   : (L*4)   LOPEN   = INPUT : .TRUE.  => INPUT DATA SET OPEN.
C                                    .FALSE. => INPUT DATA SET CLOSED.
C                            OUTPUT: ALWAYS RETURNED AS .FALSE.
C  OUTPUT: (I*4)   IRCODE  = RETURN CODE FROM SUBROUTINE:
C                            0 => NO ERROR DETECTED.
C                            2 => DISCREPANCY BETWEEN REQUESTED ELEMENT
C                                 AND THAT IN INPUT DATA FILE.
C                            3 => SELECTED DATA-BLOCK  OUT OF RANGE  OR
C                                 DOES NOT EXIST.
C
C           (I*4)   I4UNIT  = FUNCTION (SEE ROUTINE SECTION BELOW)
C
C           (C*44)  DSNAME  = FULL MVS NAME OF DATA SET OPENED
C
C ROUTINES:
C           ROUTINE    SOURCE    BRIEF DESCRIPTION
C           ------------------------------------------------------------
C           E2FILE     ADAS      OPEN DATA SET FOR SELECTED ELEMENT
C           I4UNIT     ADAS      FETCH UNIT NUMBER FOR OUTPUT OF MESSAGES
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/37
C          JET EXT. 2520
C
C DATE:    06/06/91
C
C UPDATE:  23/04/93 - PE BRIDEN - ADAS91: ADDED I4UNIT FUNCTION TO WRITE
C                                         STATEMENTS FOR SCREEN MESSAGES
C
C UPDATE:  24/05/93 - PE BRIDEN - ADAS91: CHANGED I4UNIT(0)-> I4UNIT(-1)
```

```
C
C UNIX-IDL PORT: H.P.SUMMERS
C
C VERSION: 1.1 DATE: 30-04-96
C MODIFIED: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C     - PUT UNDER SCCS CONTROL
C
C-----------------------------------------------------------------------
C
C VERSION: 1.1 DATE: 30-04-96
C MODIFIED: UNKNOWN (SOMEONE FROM TESSELLA SUPPORT SERVICES PLC)
C     - PUT UNDER SCCS CONTROL
C
C VERSION: 1.2 DATE: 20-07-07
C MODIFIED: Allan Whiteford
C     - Small modification to comments to allow for automatic
C             documentation preparation.
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IBSEL,       IRCODE,      IUNIT,       IZ0
      INTEGER             IZ0IN,       NBSEL
      LOGICAL             LOPEN
```

6.32 e9data: Subroutine e9data from library adas5xx

```
C
       SUBROUTINE E9DATA( IUNIT  , DSNAME ,
     &                    NSTORE , NEDIM  ,
     &                    ESYM   , IZ0    ,
     &                    NBSEL  , ISELA  ,
     &                    IZ     , IZ1    ,
     &                    CDONOR , CRECVR , CFSTAT , CTYPE,
     &                    ALPH0  ,
     &                    IEA    ,
     &                    TEEA   , SCX
     &                  )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: E9DATA *********************
C
C  PURPOSE:  TO  FETCH  DATA  FROM  INPUT CHARGE EXCHANGE CROSS-SECTION
C            DATA FOR  GIVEN DONOR AND RECEIVER IONS.
C
C            (MEMBER STORED IN SCX#<DONOR>.DATA (<PREFIX>.#<RECEIVER>)
C             WHERE <DONOR> ='H0' ETC. ; <RECEIVER> ='C2', 'C6' ETC.
C             AND <PREFIX. = <BLANK. OR THREE CHARACTERS
C
C  CALLING PROGRAM: ADAS509/SSCX
C
C  DATA:
C
C           UP TO 'NSTORE' SETS (DATA-BLOCKS) OF DATA MAY BE  READ FROM
C           THE FILE - EACH BLOCK FORMING A COMPLETE SET OF  CROSS-
C           SECTION VALUES FOR GIVEN COLLISION ENERGIES.
C           EACH DATA-BLOCK  IS  ANALYSED INDEPENDENTLY OF ANY  OTHER
C           DATA-BLOCK.
C
C           THE UNITS USED IN THE DATA FILE ARE TAKEN AS FOLLOWS:
C
C           COLLISION ENERGIES  : EV/AMU
C           CROSS-SECTION       : CM**2
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  IUNIT    = UNIT TO WHICH INPUT FILE IS ALLOCATED.
C  INPUT : (C*44) DSNAME   = MVS DATA SET NAME OF DATA SET BEING READ
C
C  INPUT : (I*4)  NSTORE   = MAXIMUM NUMBER  OF  INPUT DATA-BLOCKS  THAT
C                            CAN BE STORED.
C  INPUT : (I*4)  NEDIM    = MAX NUMBER OF COLLISION ENERGIES ALLOWED
C
C  OUTPUT: (C*2)  ESYM     = READ - RECEIVING ION - ELEMENT SYMBOL
C  OUTPUT: (I*4)  IZ0      = READ - RECEIVING ION - NUCLEAR CHARGE
C
C  OUTPUT: (I*4)  NBSEL    = NUMBER OF DATA-BLOCKS ACCEPTED & READ IN.
C  OUTPUT: (I*4)  ISELA()  = READ - DATA-SET DATA-BLOCK ENTRY INDICES
C                            DIMENSION: DATA-BLOCK INDEX
C
C  OUTPUT: (I*4)  IZ       = READ - RECEIVED ION -  CHARGE
C  OUTPUT: (I*4)  IZ1      = READ - RECEIVING ION - CHARGE
C
C  OUTPUT: (C*9)  CDONOR() = READ - DONOR ION IDENTIFICATION
C                            DIMENSION: DATA-BLOCK INDEX
C  OUTPUT: (C*9)  CFCODE() = READ - RECEIVER ION IDENTIFICATION
```

```
C                               DIMENSION: DATA-BLOCK INDEX
C  OUTPUT: (C*10) CFSTAT() = READ - FINAL STATE SPECIFICATION
C                               DIMENSION: DATA-BLOCK INDEX
C  OUTPUT: (C*2)  CTYPE()  = READ - CROSS-SECTION TYPE
C                               DIMENSION: DATA-BLOCK INDEX
C
C  OUTPUT: (R*8)  ALPH0()  = READ - LOW ENERGY EXTRAPOLATION PARM.
C                               DIMENSION: DATA-BLOCK INDEX
C
C
C  OUTPUT: (I*4)  IEA()    = READ - NUMBER OF COLLISION ENERGIES
C                               DIMENSION: DATA-BLOCK INDEX
C
C  OUTPUT: (R*8)  TEEA(,)  = READ - COLLISION ENERGIES (UNITS: eV/AMU)
C                               1st DIMENSION: COLLISION ENERGY INDEX
C                               2nd DIMENSION: DATA-BLOCK INDEX
C
C  OUTPUT: (R*8)  SCX(,)   =READ - FULL SET OF COLLISION CROSS-
C                                       SECTION VALUES (cm**2)
C                               1st DIMENSION: COLLISION ENERGY INDEX
C                               2nd DIMENSION: DATA-BLOCK INDEX
C
C          (I*4)  I4EIZ0   = FUNCTION - (SEE ROUTINES SECTION BELOW)
C          (I*4)  I4FCTN   = FUNCTION - (SEE ROUTINES SECTION BELOW)
C          (I*4)  I4UNIT   = FUNCTION - (SEE ROUTINE SECTION BELOW)
C          (I*4)  IBLK     = ARRAY INDEX: DATA-BLOCK INDEX
C          (I*4)  ITT      = ARRAY INDEX: COLLISION ENERGY INDEX
C          (I*4)  NENUM    = NUMBER OF COLLISION ENERGIES FOR CURRENT
C                            DATA-BLOCK
C          (I*4)  IABT     = RETURN CODE FROM 'I4FCTN'
C          (I*4)  IPOS1    = GENERAL USE STRING INDEX VARIABLE
C          (I*4)  IPOS2    = GENERAL USE STRING INDEX VARIABLE
C
C          (R*8)  R8FCTN   = FUNCTION - (SEE ROUTINES SECTION BELOW)
C
C          (L*4)  LBEND    = IDENTIFIES WHETHER THE LAST OF THE  INPUT
C                            DATA SUB-BLOCKS HAS BEEN LOCATED.
C                            (.TRUE. => END OF SUB-BLOCKS REACHED)
C
C          (C*1)  CSLASH   = '/' - DELIMITER FOR 'XXHKEY'
C          (C*2)  C2       = GENERAL USE TWO BYTE CHARACTER STRING
C          (C*3)  CKEY1    = 'FST'    - INPUT BLOCK HEADER KEY
C          (C*4)  CKEY2    = 'TYPE'   - INPUT BLOCK HEADER KEY
C          (C*5)  CKEY3    = 'ALPH0'  - INPUT BLOCK HEADER KEY
C          (C*4)  CKEY4    = 'ISEL'  - INPUT BLOCK HEADER KEY
C          (C*10) C10      = GENERAL USE TEN BYTE CHARACTER STRING
C          (C*80) C80      = GENERAL USE 80 BYTE  CHARACTER  STRING  FOR
C                            THE INPUT OF DATA-SET RECORDS.
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          XXHKEY     ADAS      OBTAIN KEY/RESPONSE STRINGS FROM TEXT
C          I4EIZ0     ADAS      INTEGER*4 FUNCTION    -
C                               RETURNS Z0 FOR GIVEN ELEMENT SYMBOL
C          I4FCTN     ADAS      INTEGER*4 FUNCTION    -
C                               CONVERT CHARACTER STRING TO INTEGER
C          I4UNIT     ADAS      INTEGER*4 FUNCTION    -
C                               FETCH UNIT NUMBER FOR OUTPUT OF MESSAGES
C          R8FCTN     ADAS      REAL*8 FUNCTION       -
C                               CONVERT CHARACTER STRING TO REAL*8
```

```
C
C AUTHOR:  H. P. SUMMERS, UNIVERSITY OF STRATHCLYDE
C          JA8.08
C          TEL. 0141-553-4196
C
C DATE:    06/06/96
C
C UNIX-IDL PORT: H.P.SUMMERS
C
C VERSION: 1.1 DATE: 30-04-96
C MODIFIED: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C     - PUT UNDER SCCS CONTROL
C
C VERSION: 1.2 DATE: 20-07-07
C MODIFIED: Allan Whiteford
C     - Small modification to comments to allow for automatic
C             documentation preparation.
C
C-----------------------------------------------------------------------
      CHARACTER*9         CDONOR(NSTORE)
      CHARACTER*10        CFSTAT(NSTORE)
      CHARACTER*9         CRECVR(NSTORE)
      CHARACTER*2         CTYPE(NSTORE)
      CHARACTER*44        DSNAME
      CHARACTER*2         ESYM
      INTEGER             IEA(NSTORE), ISELA(NSTORE),            IUNIT
      INTEGER             IZ,          IZ0,         IZ1,         NBSEL
      INTEGER             NEDIM,       NSTORE
      REAL*8              ALPH0(NSTORE),            SCX(NEDIM,NSTORE)
      REAL*8              TEEA(NEDIM,NSTORE)
```

6.33 e9econ: Subroutine e9econ from library adas5xx

```
C
      SUBROUTINE E9ECON( INTYP, OUTTYP, AMD,AMR,IEVAL, EIN, EOUT )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: E9ECON *********************
C
C  PURPOSE: TO CONVERT AN ARRAY OF COLLISION  ENERGIES INTO A  SPECIFIED
C           FORM.
C
C  CALLING PROGRAM: GENERAL USE
C
C  SUBROUTINE:
C
C  INPUT :    (I*4)  INTYP   = 1 => 'EIN(ARRAY)' UNITS: DONOR   EV
C                              = 2 => 'EIN(ARRAY)' UNITS: RECVR   EV
C                              = 3 => 'EIN(ARRAY)' UNITS: ENERGY  EV/AMU
C  INPUT :    (I*4)  OUTTYP  = 1 => 'EOUT(ARRAY)' UNITS: DONOR  EV
C                              = 2 => 'EOUT(ARRAY)' UNITS: RECVR  EV
C                              = 3 => 'EOUT(ARRAY)' UNITS: ENERGY EV/AMU
C  INPUT :    (R*8)  AMD     = DONOR MASS NUMBER
C  INPUT :    (R*8)  AMR     = RECEIVER MASS NUMBER
C  INPUT :    (I*4)  IEVAL   = NO. OF ENERGIES IN EIN(ARRAY)
C  INPUT :    (R*8)  EIN()   = INPUT  ENERGIES (STATED UNITS)
C  OUTPUT:    (R*8)  EOUT()  = OUTPUT ENERGIES (STATED UNITS)
C
C
C             (I*4)  I       = GENERAL USE
C
C             (R*8)  ECONV() = ENERGY/VELOCITY CONVERSION PARAMETERS
C
C ROUTINES:  NONE
C
C
C AUTHOR:  H. P. SUMMERS, UNIVERSITY OF STRATHCLYDE
C          JA8.08
C          TEL.  0141-553-4196
C
C DATE:    17/11/95
C
C UNIX-IDL PORT: H.P.SUMMERS
C
C VERSION: 1.1 DATE: 30-04-96
C MODIFIED: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C     - PUT UNDER SCCS CONTROL
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IEVAL,       INTYP,       OUTTYP
      REAL*8              AMD,         AMR,         EIN(IEVAL)
      REAL*8              EOUT(IEVAL)
```

6.34 e9spln: Subroutine e9spln from library adas5xx

```
C
      SUBROUTINE E9SPLN( NEDIM  ,
     &                   IEA    , IEVAL   ,
     &                   TEEA   , EEVA    ,
     &                   SCX    , SCXA    ,
     &                   LERNG
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: E9SPLN *********************
C
C  PURPOSE:
C          PERFORMS CUBIC SPLINE ON LOG(ENERGY <EV/AMU> ) VERSUS
C          LOG(CX CROSS-SECTION <CM2>).
C          INPUT DATA FOR A GIVEN DONOR/RECEIVER COMBINATION DATA-BLOCK.
C
C          USING  ONE-WAY SPLINES  IT  CALCULATES  THE  CROSS-SECTIONS
C          FOR  'IEVAL'  COLLISION ENERGIES VALUES  FROM
C          THE LIST OF COLLISION ENERGIES READ IN FROM THE INPUT FILE
C
C          IF A  VALUE  CANNOT  BE  INTERPOLATED  USING  SPLINES  IT  IS
C          EXTRAPOLATED VIA 'XXSPLE'. (SEE NOTES BELOW).
C
C  CALLING PROGRAM: ADAS509/SSCX
C
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  IEA     = INPUT DATA FILE: NUMBER OF COLLISION ENER-
C                           GIES READ FOR THE DATA-BLOCK BEING ASSESSED
C  INPUT : (I*4)  IEVAL   = NUMBER OF ISPF ENTERED COLLISION ENERGIES
C                           VALUES FOR  WHICH  CX CROSS-SECTIONS
C                           ARE REQUIRED FOR TABULAR/GRAPHICAL OUTPUT.
C
C  INPUT : (R*8)  TEEA()  = INPUT DATA FILE: COLLISION ENERGIES (EV/AMU)
C                           FOR THE DATA-BLOCK BEING ASSESSED.
C                           DIMENSION: COLLISION ENERGY INDEX
C  INPUT : (R*8)  EEVA()  = USER ENTERED: COLLISION ENERGIES (EV/AMU)
C                           DIMENSION: COLLISION ENERGY INDEX
C
C  INPUT : (R*8)  SCX()    =INPUT DATA FILE: FULL SET OF CX CROSS-
C                           SECTIONS FOR THE DATA-BLOCK BEING ANALYSED
C                           1ST DIMENSION: COLLISION ENERGY INDEX
C  OUTPUT: (R*8)  SCXA()  = SPLINE INTERPOLATED OR  EXTRAPOLATED
C                           CX CROSS-SECTIONS FOR
C                           THE USER ENTERED COLLISION ENERGIES.
C                           DIMENSION: COLLISION ENERGIES INDEX
C
C  OUTPUT: (L*4)  LERNG()=  .TRUE.  => OUTPUT 'SCXA()' VALUE WAS INTER-
C                                      POLATED  FOR  THE  USER  ENTERED
C                                      COLLISION ENERGY 'EEVA()'.
C                           .FALSE. => OUTPUT 'SCXA()' VALUE WAS EXTRA-
C                                      POLATED  FOR  THE  USER  ENTERED
C                                      COLLISION ENERGY 'EEVA()'.
C                           DIMENSION: COLLISION ENERGY INDEX
C
C          (I*4)  NIN     = PARAMETER = MAX. NO. OF INPUT  ENERGY
C                                       VALUES. MUST BE >= 'IEA'
C          (I*4)  NOUT    = PARAMETER = MAX. NO. OF OUTPUT ENERGY
```

```
C                                       VALUES.  MUST BE >= 'IEVAL'
C          (I*4)  L1      = PARAMETER = 1
C
C
C          (I*4)  IET     = ARRAY SUBSCRIPT USED INPUT  FILE  COLLISION
C                           ENERGIES.
C          (I*4)  IT      = ARRAY  SUBSCRIPT  USED  FOR  USER  ENTERED
C                           COLLISION ENERGIES.
C          (I*4)  IOPT    = DEFINES THE BOUNDARY DERIVATIVES FOR THE
C                           SPLINE ROUTINE 'XXSPLE', SEE 'XXSPLE'.
C                           (VALID VALUES = <0, 0, 1, 2, 3, 4)
C
C          (L*4)  LSETX   = .TRUE.  => SET UP SPLINE PARAMETERS RELATING
C                                      TO 'XIN' AXIS.
C                           .FALSE. => DO NOT SET UP SPLINE PARAMETERS
C                                      RELATING TO 'XIN' AXIS.
C                                      (I.E. THEY WERE SET IN A PREVIOUS
C                                            CALL )
C                           (VALUE SET TO .FALSE. BY 'XXSPLE')
C
C          (R*8)  R8FUN1  = FUNCTION - (SEE ROUTINES SECTION BELOW)
C
C          (R*8)  XIN()   = LOG( DATA FILE COLLISION ENERGIES )
C          (R*8)  YIN()   = LOG( DATA FILE CX CROSS-SECTIONS)
C          (R*8)  XOUT()  = LOG( USER ENTERED COLLISION ENERGIES.)
C          (R*8)  YOUT()  = LOG( OUTPUT GENERATED CX CROSS-SECTIONS)
C          (R*8)  DF()    = SPLINE INTERPOLATED DERIVATIVES
C
C
C NOTE:
C
C          ONE-DIMENSIONAL SPLINE CARRIED OUT BY THIS SUBROUTINE:
C
C           LOG( SCX )  vs. LOG( E )
C
C           E      = COLLISION ENERGY (units: eV/AMU)
C           SCX    = CX CROSS-SECTION (units: cm**2)
C
C           Extrapolation criteria:
C
C           Low  E: zero gradient  extrapolation (i.e. DY(1)  = 0.0)
C           High E: zero curvature extrapolation (i.e. DDY(N) = 0.0)
C
C           (These criteria are met by calling XXSPLE with IOPT=4)
C
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          XXSPLE     ADAS      SPLINE SUBROUTINE (EXTENDED DIAGNOSTICS)
C          R8FUN1     ADAS      REAL*8 FUNCTION: ( X -> X )
C
C AUTHOR:  H. P. SUMMERS, UNIVERSITY OF STRATHCLYDE
C          JA8.08
C          TEL. 0141-553-4196
C
C DATE:    16/11/95
C
C UNIX-IDL PORT: H.P.SUMMERS
C
C VERSION: 1.1 DATE: 30-04-96
```

```
C MODIFIED: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C     - PUT UNDER SCCS CONTROL
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IEA,         IEVAL,       NEDIM
      LOGICAL             LERNG(IEVAL)
      REAL*8              EEVA(IEVAL), SCX(NEDIM),  SCXA(IEVAL)
      REAL*8              TEEA(NEDIM)
```

6.35 e9titl: Subroutine e9titl from library adas5xx

```
C
      SUBROUTINE E9TITL( IBSEL  , DSFULL ,
     &                   CDONOR , CRECVR ,
     &                   CFSTAT , CTYPE ,
     &                   TITLX
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: E9TITL *********************
C
C  PURPOSE:  TO CREATE THE DESCRIPTIVE TITLE FOR SELECTED DATA-BLOCK.
C
C  CALLING PROGRAM: ADAS509/SSCX
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  IBSEL    = SELECTED DATA-BLOCK: INDEX
C  INPUT : (C*80) DSFULL   = FULL MVS INPUT DATA SET NAME
C
C  INPUT : (C*9)  CDONOR   = SELECTED DATA-BLOCK: DONOR
C                                                 IDENTIFICATION.
C  INPUT : (C*9)  CRECVR   = SELECTED DATA-BLOCK: RECEIVER
C                                                 IDENTIFICATION.
C
C  INPUT : (C*10) CFSTAT   = SELECTED DATA-BLOCK: FINAL STATE
C                                                 SPECIFICATION
C  INPUT : (C*2)  CTYPE    = SELECTED DATA-BLOCK: CROSS. SECT. TYPE
C
C  OUTPUT: (C*120) TITLX    = SELECTED DATA-BLOCK: DESCRIPTIVE TITLE
C
C          (C*2)  C2       = GENERAL USE 2 BYTE  CHARACTER  STRING
C    (I*4)  POS_NOW  = CURRENT POSITION IN TITLE STRING
C    (I*4)  LEN_NAME = LENGTH OF FILENAME
C          (I*4)  IFIRST   = POSITION OF FIRST CHARACTER IN FILENAME
C          (I*4)  ILAST    = POSITION OF LAST CHARACTER IN FILENAME
C
C ROUTINES:
C          XXSLEN          = UTILITY ROUTINE WHICH FINDS FIRST AND LAST
C                            NON-BLANK CHARACTERS IN A STRING.
C
C AUTHOR:  H. P. SUMMERS
C          JA8.08
C          TEL. 0141-553-4196
C
C DATE:    11/03/96
C
C UNIX-IDL PORT: H.P.SUMMERS
C
C VERSION: 1.1 DATE: 30-04-96
C MODIFIED: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C     - PUT UNDER SCCS CONTROL
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      CHARACTER*9         CDONOR
      CHARACTER*10        CFSTAT
      CHARACTER*9         CRECVR
      CHARACTER*2         CTYPE
      CHARACTER*80        DSFULL
```

```
CHARACTER*120       TITLX
INTEGER             IBSEL
```

6.36 e9vrdc: Subroutine e9vrdc from library adas5xx

```
C
      SUBROUTINE E9VRDC( ICIND  ,
     &                   NBSEL  ,
     &                   CDONOR , CRECVR, CFSTAT , CTYPE ,
     &                   SCI    ,
     &                   SDONOR , SRECVR, SFSTAT , STYPE
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: E9VRDC *********************
C
C  PURPOSE: TO SETUP DONOR/RECVR CHARACTER STRINGS FOR USE WITH SETTING
C           ISPF VARIABLES WHEN  SELECTED CROSS-SECTIONS ARE DISPLAYED.
C
C  CALLING PROGRAM: E9PAN2
C
C  SUBROUTINE:
C
C  INPUT : (I*4)   ICIND    = SELECTED/REQUESTED CROSS-SECTION DATA-BLOCK
C                             INDEX.
C
C  INPUT : (I*4)   NBSEL    = NUMBER OF DATA-BLOCK CROSS-SECTIONS.
C
C  INPUT : (C*9)   CDONOR() = INPUT DATA FILE: DONOR ION IDENTIFICATION
C                             DIMENSION: DATA-BLOCK INDEX
C  INPUT : (C*9)   CRECVR() = INPUT DATA FILE: RECEIVER ION IDENTIFICATION
C                             DIMENSION: DATA-BLOCK INDEX
C  INPUT : (C*10)  CFSTAT() = INPUT DATA FILE: FINAL STATE SPECIFICATION
C                             DIMENSION: DATA-BLOCK INDEX
C  INPUT : (C*2)   CTYPE()  = INPUT DATA FILE: CROSS-SECTION TYPE
C                             DIMENSION: DATA-BLOCK INDEX
C                             DIMENSION: DATA-BLOCK INDEX
C
C  OUTPUT: (C*3)   SCI      = ISPF PANEL VARIABLE: DATA-BLOCK INDEX
C
C  OUTPUT: (C*9)   SDONOR   = ISPF PANEL VARIABLE: DONOR ION IDENTIFICATION
C  OUTPUT: (C*9)   SRECVR   = ISPF PANEL VARIABLE: RECEIVER ION IDENTIFICATION
C  OUTPUT: (C*10)  SFSTAT   = ISPF PANEL VARIABLE: FINAL STATE SPECIFICATION
C  OUTPUT: (C*2)   STYPE    = ISPF PANEL VARIABLE: CROSS-SECTION TYPE
C
C
C ROUTINES: NONE
C
C AUTHOR:  H. P. SUMMERS, UNIVERSITY OF STRATHCLYDE
C          JA8.08
C          TEL.  0141-553-4196
C
C DATE:    15/11/95
C
C UNIX-IDL PORT: H.P.SUMMERS
C
C VERSION: 1.1 DATE: 30-04-96
C MODIFIED: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C     - PUT UNDER SCCS CONTROL
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
```

```
CHARACTER*9         CDONOR(NBSEL)
CHARACTER*10        CFSTAT(NBSEL)
CHARACTER*9         CRECVR(NBSEL)
CHARACTER*2         CTYPE(NBSEL)
CHARACTER*3         SCI
CHARACTER*9         SDONOR
CHARACTER*10        SFSTAT
CHARACTER*9         SRECVR
CHARACTER*2         STYPE
INTEGER             ICIND,       NBSEL
```

6.37 exther: Subroutine exther from library adas5xx

```
C
       SUBROUTINE EXTHER ( NEDIM  , NTDIM  ,
     &                     LSETX  , LPASS  ,
     &                     AMDON  , AMREC  , ILTYP  ,
     &                     NENIN  , ENIN   , NENOUT , ENOUT ,
     &                     SGIN   , RCOUT
     &                   )
C-----------------------------------------------------------------------
C
C  ********************** FORTRAN77 SUBROUTINE: EXTHER *****************
C
C  VERSION: 1.0 (ADAS91)
C
C  PURPOSE:  OBTAINS RATE COEFFICIENTS FOR DONOR/RECEIVER CHARGE
C            EXCHANGE COLLISIONS FOR CASES OF THERMAL DONOR AND
C            THERMAL RECEIVER FROM CROSS-SECTION TABULATIONS. AN ARRAY
C            OF VALUES IS PRODUCED.
C
C  CALLING PROGRAM:  ADAS509
C
C  SUBROUTINE:
C
C  INPUT :  (I*4)  NEDIM    = MAX. NUMBER OF ENERGIES IN SOURCE DATA
C                             VECTOR
C  INPUT :  (I*4)  NTDIM    = MAX. NUMBER OF TEMPERATURES IN OUTPUT
C                             VECTOR
C  INPUT :  (L*4)  LSETX    = .TRUE. => SPLINE PRESET FOR THESE KNOTS
C                             .FLSE. => SPLINE NOT SET FOR THESE KNOTS
C  INPUT :  (L*4)  LPASS    = .TRUE. => DO NOT CONVERT INTO LOG10 FOR
C                                       ENERGIES AND X-SECTS. FOR SPLINE
C                             .FLSE. => CONVERT INTO LOG10 FOR
C                                       ENERGIES AND X-SECTS. FOR SPLINE
C  INPUT :  (R*8)  AMDON    = DONOR MASS NUMBER
C  INPUT :  (R*8)  AMREC    = RECEIVER MASS NUMBER
C  INPUT : (I*4)   ILTYP    = TYPE FOR LOW AND HIGH ENERGY CROSS-
C                             SECTION EXTRAPOLATION
C
C  INPUT : (I*4)   NENIN    = NUMBER OF ENERGIES IN INPUT DATA SET
C  INPUT : (R*8)   ENIN()   = ENERGIES (EV/AMU) IN INPUT DATA SET
C  INPUT : (I*4)   NENOUT   = NUMBER OF TEMPERATURES FOR OUTPUT DATA SET
C  INPUT : (R*8)   ENOUT()  = TEMPERATURES (EV) FOR OUTPUT DATA SET
C  INPUT : (R*8)   SGIN()   = INPUT X-SECTIONS (CM2) FROM INPUT DATA SET
C                             1ST.DIM: ENERGY INDEX
C  OUTPUT: (R*8)   RCOUT(,) = RATE COEFF. (CM3 S-1) IN OUTPUT DATA SET
C                             1ST.DIM: DONOR TEMPERATURE INDEX
C                             2ND.DIM: RECEIVER TEMPERATURE INDEX
C
C          (I*4)   I        = GENERAL INDEX
C          (I*4)   IT       = GENERAL INDEX
C          (I*4)   ITR      = GENERAL INDEX
C          (I*4)   ITD      = GENERAL INDEX
C          (I*4)   ITHETA   = GENERAL INDEX
C          (I*4)   IOPT     = SPLINE END POINT CURVATURE/GRADIENT OPTION
C                             1 => DDY1 = 0, DDYN = 0
C                             4 => DY1 = 0 , DDYN = 0
C          (I*4)   IXD      = DONOR GAUSSIAN QUADRATURE INDEX
C          (I*4)   IXR      = RECEIVER GAUSSIAN QUADRATURE INDEX
C          (I*4)   NGS      = GAUSSIAN QUARATURE DIMENSION
C          (I*4)   NTHETA   = NUMBER OF ANGLE VALUES FOR QUADRATURE
```

```
C          (I*4)   LTHETA   = NTHETA+1
C          (I*4)   L1       = PARAMETER = 1
C
C          (R*8)   ETHD     = THERMAL ENERGY OF DONOR       (JOULES)
C          (R*8)   ETHR     = THERMAL ENERGY RECEIVER       (JOULES)
C          (R*8)   HSIMP    = SIMPSON'S RULE STEP INTERVAL
C          (R*8)   THETA    = ANGLE BETWEEN PARTICLE VELOCITIES (RAD)
C          (R*8)   FAC      = GENERAL VARIABLE
C          (R*8)   FLAG     = GENERAL VARIABLE
C          (R*8)   XMDKG    = DONOR MASS    (KG)
C          (R*8)   XMRKG    = RECEIVER MASS (KG)
C          (R*8)   VD       = DONOR SPEED    (M S-1)
C          (R*8)   VR       = RECEIVER SPEED (M S-1)
C          (R*8)   RATE     = EVALUATED RATE COEFFICIENT (CM3 S-1)
C          (R*8)   PART1    = GENERAL VARIABLE
C          (R*8)   PART2    = GENERAL VARIABLE
C          (R*8)   PART3    = GENERAL VARIABLE
C          (R*8)   PART12   = GENERAL VARIABLE
C          (R*8)   PART23   = GENERAL VARIABLE
C          (R*8)   PART123  = GENERAL VARIABLE
C          (R*8)   VREL1    = GENERAL RELATIVE SPEED VARIABLE
C          (R*8)   XSEC1    = GENERAL CROSS-SECTION VARIABLE
C          (R*8)   VAL      = GENERAL VARIABLE
C
C          (R*8)   XGS()    = GAUSSIAN QUADRATURE NODES
C          (R*8)   WGS()    = GAUSSIAN QUADRATURE WEIGHTS
C          (R*8)   VREL()   = RELATIVE SPEED OF PARTICLES FOR DIFFERENT
C                             ANGLES (CM S-1)
C          (R*8)   XSEC()   = CHARGE EXCHANGE CROSS-SECTIONS FOR
C                             RELATIVE SPEEDS AT DIFFERENT ANGLES (CM2)
C
C
C  ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          SIGCX      ADAS      INTERPOLATES CX CROSS-SECTION TABLES
C
C
C  AUTHOR:  H. P. SUMMERS, UNIVERSITY OF STRATHCLYDE
C           JA8.08
C           TEL. 0141-553-4196
C
C  DATE:    17/11/95
C
C UNIX-IDL PORT: H.P.SUMMERS
C
C VERSION: 1.1 DATE: 30-04-96
C MODIFIED: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C     - PUT UNDER SCCS CONTROL
C
C VERSION : 1.2
C DATE    : 15-12-2006
C MODIFIED: Martin O'Mullane
C           - Trap vrel=0 before passing to sigcx.
C
C-----------------------------------------------------------------------
      INTEGER             ILTYP,       NEDIM,       NENIN,       NENOUT
      INTEGER             NTDIM
      LOGICAL             LPASS,       LSETX
      REAL*8              AMDON,       AMREC,       ENIN(NEDIM)
      REAL*8              ENOUT(NTDIM),             RCOUT(NTDIM,NTDIM)
```

```
      REAL*8              SGIN(NEDIM)
```

6.38 sigcx: Subroutine sigcx from library adas5xx

```
C
      SUBROUTINE SIGCX ( LSETX    , LPASS    , ILTYP    , IOPT     ,
     &                   NENIN    , ENIN     , SGIN     ,
     &                   LTHETA   , VREL     , XSEC
     &                 )
C-----------------------------------------------------------------------
C
C  ********************** FORTRAN77 SUBROUTINE: SIGCX *****************
C
C  VERSION: 1.0 (ADAS91)
C
C  PURPOSE:  INTERPOLATES CROSS-SECTION DATA FROM AN INPUT VECTOR OF
C            VALUES USING CUBIC SPLINES.
C
C            EXTRAPOLATES FOR RELATIVE SPEEDS OUT OF DATA RANGE
C            ACCORDING TO VARIOUS TYPES (ILTYP).  LOGARITHMIC
C            INTERPOLATION MAY BE USED (LPASS). SPEED ECONOMY IS
C            POSSIBLE FOR REPEATS WITH THE SAME SPLINE KNOTS (LSETX).
C
C  CALLING PROGRAM:  CXTHER
C
C  NOTES:
C        (1) FOR  ILTYP.EQ.0, EXTRAPOLATION IS AS FOLLOWS:
C              XSEC = SIG0*DEXP(-ALPH0/VREL) FOR VREL<VREL(MIN)
C              XSEC = SIG1*VREL**(-7.0)  FOR VREL> VREL(MAX),
C              WHERE VREL(MIN), VREL(MAX) ARE THE FIRST AND LAST         FROM
C              INPUT VALUES IN DATA TABLES IN ADF24.
C            FOR  ILTYP.NE.0, EXTRAPOLATION IS AS AS ABOVE AT THIS
C              TIME.
C
C
C  SUBROUTINE:
C
C  INPUT : (L*4)   LSETX    = .TRUE. => SPLINE NOT SET FOR THESE KNOTS
C                             .FLSE. => SPLINE NOT FOR THESE KNOTS
C  INPUT : (L*4)   LPASS    = .TRUE. => DO NOT CONVERT INTO LOG10 FOR
C                                       ENERGIES AND X-SECTS. FOR SPLINE
C                             .FLSE. => CONVERT INTO LOG10 FOR
C                                       ENERGIES AND X-SECTS. FOR SPLINE
C  INPUT : (I*4)   ILTYP    = TYPE FOR LOW AND HIGH ENERGY CROSS-
C                             -SECTION EXTRAPOLATION.
C  INPUT : (I*4)   IOPT     = SPLINE END POINT CURVATURE/GRADIENT OPTION
C                             1 => DDY1 = 0, DDYN = 0
C                             4 => DY1 = 0 , DDYN = 0
C
C  INPUT : (I*4)   NENIN    = NUMBER OF ENERGIES IN INPUT DATA SET
C  INPUT : (R*8)   ENIN()   = ENERGIES (EV/AMU) IN INPUT DATA SET
C  INPUT : (R*8)   SGIN()   = INPUT X-SECTIONS (CM2) FROM INPUT DATA SET
C                             1ST.DIM: ENERGY INDEX
C  INPUT : (I*4)   LTHETA   = NUMBER OF VALUES IN VREL VECTOR
C  INPUT : (R*8)   VREL()   = RELATIVE SPEEDS FOR OUTPUT (CM S-1)
C
C  OUTPUT: (R*8)   XSEC()   = OUTPUT CROSS-SECTION (CM2)
C
C          (I*4)   MAXENS   = PARAMETER = MAX. LENGTH OF TABULAR XSECT.
C                                         VECTOR
C          (I*4)   LDTHET   = PARAMETER = MAX. LENGTH OF INTERNAL
C                                         VECTORS
C          (R*8)   CMSAMU   = PARAMETER = CONVERSION FACTOR FOR ENERGY
```

```
C                                        (AMU) TO VELOCITY (CM S-1)
C
C          (I*4)   I        = GENERAL INDEX
C          (I*4)   N        = GENERAL INDEX
C          (R*8)   ALPH0    = LOW VELOCITY EXTRAPOLATION PARAMETER
C          (R*8)   EXPON    = EXPONENT OF EXPONENTIAL
C          (R*8)   VSLOPE   = HIGH VELOCITY EXTRAPOLATION PARAMETER
C          (R*8)   XIN()    = INTERNAL SPLINE INDEPENDENT VARIABLE
C          (R*8)   YIN()    = INTERNAL SPLINE DEPENDENT VARIABLE
C          (R*8)   VIN()    = INTERNAL VECTOR
C          (R*8)   DY()     = DERIVATIVES AT SPLINE KNOTS
C          (R*8)   XOUT()   = INTERNAL OUTPUT INDEPENDENT VARIABLE
C          (R*8)   YOUT()   = INTERNAL OUTPUT DEPENDENT VARIABLE
C          (L*4)   LINTRP() = .TRUE.  => POINT INTERPOLATED
C                           = .FALSE. => POINT EXTRAPOLATED
C
C
C  ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          XXSPLE     ADAS      INTERPOLATES USING CUBIC SPLINES
C          R8FUN1     ADAS      EXTERNAL FUNCTION FOR XXSPLE
C
C
C  AUTHOR:  H. P. SUMMERS, UNIVERSITY OF STRATHCLYDE
C           JA8.08
C           TEL. 0141-553-4196
C
C  DATE:    03/11/95
C
C  UPDATE:  11/04/96  HP SUMMERS - TRAPPED CASE OF ZERO RELATIVE SPEED
C
C UNIX-IDL PORT: H.P.SUMMERS
C
C VERSION: 1.1 DATE: 30-04-96
C MODIFIED: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C     - PUT UNDER SCCS CONTROL
C
C VERSION: 1.2 DATE: 30-04-96
C MODIFIED: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C     - REPLACED FINTX WITH R8FUN1
C
C VERSION: 1.3 DATE: 26-11-97
C MODIFIED: Martin O'Mullane
C           - remove extrapolation at low and high energies. We will
C             accept the results from XXSPLE (with IOPT=4) until
C             better ideas of how to deal with threshold and high energy
C             extensions are found.
C
C VERSION: 1.4 DATE: 17-05-07
C MODIFIED: Allan Whiteford
C           - Updated comments as part of subroutine documentation
C             procedure.
C
C-----------------------------------------------------------------------
      INTEGER             ILTYP,       IOPT,        LTHETA,      NENIN
      LOGICAL             LPASS,       LSETX
      REAL*8              ENIN(NENIN), SGIN(NENIN), VREL(LTHETA)
      REAL*8              XSEC(LTHETA)
```

6.39 spec: Subroutine spec from library adas5xx

```
      SUBROUTINE SPEC( IBSEL  , IZIN   , IZ0IN  ,
     &                 ITVAL  , TVAL   , DVAL   ,
     &                 WLNGTH ,
     &                 PECA   , LTRNG  , LDRNG  ,
     &                 TITLX  , IRCODE
     &               )
C-----------------------------------------------------------------------
C
C  ******************* FORTRAN77 SUBROUTINE: SPEC **********************
C
C  PURPOSE: TO EXTRACT AND INTERPOLATE  PHOTON EMISSIVITIES FOR
C           EMITTING IONS.
C           USES THE SAME ROUTINES USED BY ADAS503, EXCEPT FOR:
C
C           'E3FILE' - WHICH OPENS THE REQUESTED FILE.
C           'E3CHKB' - WHICH CHECKS INPUT  VALUES  ARE  CONSISTENT  WITH
C                      THE SELECTED DATA-BLOCK 'IBSEL' AND   'IBSEL'  IS
C                      IN RANGE.
C
C            THE FIRST OF THESE FUNCTIONS IS CARRIED  OUT  IN  'ADAS503'
C            VIA ISPF PANELS USING THE ROUTINE 'E3SPF0'  -  ADAS503 DOES
C            NOT REQUIRE THE ROUTINE 'E3CHKB' AS THE USER CANNOT  SELECT
C            AN INVALID VALUE FOR 'IBSEL' OR 'IBSEL'/EMITTER COMBINATION
C
C
C  CALLING PROGRAM: GENERAL USE
C
C  SUBROUTINE:
C
C  INPUT : (I*4)   IBSEL   = INDEX OF DATA-BLOCK SELECTED FOR ANALYSIS
C  INPUT : (I*4)   IZIN    = ION CHARGE OF EMITTING ION
C  INPUT : (I*4)   IZ0IN   = NUCLEAR CHARGE OF EMITTING ION
C
C  INPUT : (I*4)   ITVAL   = NO. OF ELECTRON TEMPERATURE/DENSITY PAIRS
C  INPUT : (R*8)   TVAL()  = ELECTRON TEMPERATURES (UNITS: EV)
C                            DIMENSION: TEMPERATURE/DENSITY PAIR INDEX
C  INPUT : (R*8)   DVAL()  = ELECTRON DENSITIES (UNITS: CM-3)
C                            DIMENSION: TEMPERATURE/DENSITY PAIR INDEX
C
C  OUTPUT: (R*8)   WLNGTH  = SELECTED BLOCK WAVELENGTH (ANGSTROMS)
C
C  OUTPUT: (R*8)   PECA()  = PHOTON EMISSIVITIES.
C                            DIMENSION: TEMPERATURE/DENSITY PAIR INDEX
C  OUTPUT: (L*4)   LTRNG() =.TRUE.  => OUTPUT 'PECA()'  VALUE WAS INTER-
C                                      POLATED  FOR  THE  USER  ENTERED
C                                      ELECTRON TEMPERATURE 'TVAL()'.
C                           .FALSE. => OUTPUT 'PECA()'  VALUE WAS EXTRA-
C                                      POLATED  FOR  THE  USER  ENTERED
C                                      ELECTRON TEMPERATURE 'TVAL()'.
C                            DIMENSION: TEMPERATURE/DENSITY PAIR INDEX
C  OUTPUT: (L*4)   LDRNG() =.TRUE.  => OUTPUT 'PECA()'  VALUE WAS INTER-
C                                      POLATED  FOR  THE  USER  ENTERED
C                                      ELECTRON DENSITY 'DVAL()'.
C                           .FALSE. => OUTPUT 'PECA()'  VALUE WAS EXTRA-
C                                      POLATED  FOR  THE  USER  ENTERED
C                                      ELECTRON DENSITY 'DVAL()'.
C                            DIMENSION: TEMPERATURE/DENSITY PAIR INDEX
C
C  OUTPUT: (C*80)  TITLX   = INFORMATION STRING (DSN ETC.)
```

```
C  OUTPUT: (I*4)   IRCODE  = RETURN CODE FROM SUBROUTINE:
C                            0 => NORMAL COMPLETION - NO ERROR DETECTED
C                            1 => DATA SET MEMBER FOR EMITTING ION WITH
C                                 CHARGE 'IZIN' & ION CHARGE 'IZ0IN' CAN
C                                 NOT BE FOUND/DOES NOT EXIST.
C                            2 => DISCREPANCY BETWEEN REQUESTED CHARGES
C                                 AND THOSE IN INPUT FILE.
C                            3 => THE SELECTED DATA-BLOCK 'IBSEL' IS OUT
C                                 OF RANGE OR DOES NOT EXIST.
C                            4 => INVALID VALUE FOR 'IZ0IN' ENTERED.
C                                 ('IZ0MIN' <= 'IZ0IN' <= 'IZ0MAX')
C                            5 => INVALID VALUE FOR 'IZIN' ENTERED.
C                                 ( 0  <= 'IZIN' <= 99 )
C                            9 => ERROR ENCOUNTERED WHEN TRYING TO OPEN
C                                 INPUT DATA-SET.
C
C           (I*4)   NSTORE  = PARAMETER= MAXIMUM NUMBER  OF  DATA-BLOCKS
C                                        WHICH CAN BE READ FROM THE INPUT
C                                        DATA-SET.
C           (I*4)   NTDIM   = PARAMETER= MAXIMUM NUMBER OF ELECTRON TEMP-
C                                        ERATURES THAT CAN BE READ  FROM
C                                        AN INPUT DATA-SET DATA-BLOCK.
C           (I*4)   NDDIM   = PARAMETER= MAXIMUM NUMBER OF ELECTRON DENS-
C                                        ITIES  THAT  CAN  BE  READ  FROM
C                                        AN INPUT DATA-SET DATA-BLOCK.
C           (I*4)   IZ0MIN  = PARAMETER: MIN. ALLOWED VALUE FOR 'IZ0IN'
C           (I*4)   IZ0MAX  = PARAMETER: MAX. ALLOWED VALUE FOR 'IZ0IN'
C
C           (I*4)   IZ0LST  = LAST VALUE OF 'IZ0IN' FOR  WHICH  INPUT
C                             DATA WAS READ.
C           (I*4)   IZLAST  = LAST VALUE OF 'IZIN' FOR  WHICH  INPUT
C                             DATA WAS READ.
C           (I*4)   IUNIT   = UNIT TO WHICH INPUT DATA SET IS ALLOCATED
C           (I*4)   NBSEL   = TOTAL NUMBER OF DATA-BLOCKS READ FROM INPUT
C                             DATA SET.
C           (I*4)   IZ0     = INPUT FILE - EMITTING ION - NUCLEAR CHARGE
C           (I*4)   IZ      = INPUT FILE - EMITTING ION - CHARGE
C           (I*4)   IZ1     = INPUT FILE - EMITTING ION - CHARGE + 1
C           (I*4)   IPOS    = USED IN CONVERTING CWAVEL -> WLNGTH
C
C           (L*4)   LOPEN   = .TRUE.  => INPUT DATA SET OPEN.
C                             .FALSE. => INPUT DATA SET CLOSED.
C
C           (C*2)   ESYM    = INPUT FILE - EMITTING ION - ELEMENT SYMBOL
C           (C*3)   EXTIN   = CURRENT ADAS SOURCE DATA FILE EXTENSION
C           (C*3)   EXTLST  = ADAS SOURCE DATA FILE EXT. USED LAST TIME
C                             DATA WAS READ.
CA          (C*80)  UIDIN   = CURRENT ADAS SOURCE DATA USER ID.
CA          (C*80)  UIDLST  = ADAS SOURCE DATA USER ID USED LAST TIME
C                             DATA WAS READ.
C           (C*8)   GRPIN   = CURRENT ADAS SOURCE DATA GROUPNAME
C           (C*8)   GRPLST  = ADAS SOURCE DATA GROUPNAME USED LAST TIME
C                             DATA WAS READ.
CA          (C*80)  TYPIN   = ADAS DATA FILE SUBDIRECTORY (OPTIONAL)
CA          (C*80)  TYPLST  = ADAS DATA FILE SUBDIRECTORY USED LAST TIME
C                             DATA WAS READ.
C           (C*11)  C11     = USED IN CONVERTING CWAVEL -> WLNGTH
CA          (C*80)  DSNREQ  = NAME OF DATA SET REQUESTED
C                             (MAY OR MAY NOT EXIST)
CA          (C*80)  DSNAME  = NAME OF DATA SET INTERROGATED
C
```

```
C          (I*4)   ISELA() = INPUT DATA FILE: DATA-BLOCK ENTRY INDICES.
C                            DIMENSION: DATA-BLOCK INDEX
C          (I*4)   ITA()   = INPUT DATA SET-NUMBER OF ELECTRON TEMPERA-
C                            TURES.
C                            DIMENSION: DATA-BLOCK INDEX
C          (I*4)   IDA()   = INPUT DATA SET-NUMBER OF ELECTRON DENSITIES
C                            DIMENSION: DATA-BLOCK INDEX
C
C          (R*8)   TETA(,) = INPUT DATA SET -
C                            ELECTRON TEMPERATURES (UNITS: eV)
C                            1st DIMENSION: ELECTRON TEMPERATURE INDEX
C                            2nd DIMENSION: DATA-BLOCK INDEX
C          (R*8)   TEDA(,) = INPUT DATA SET -
C                            ELECTRON DENSITIES    (UNITS: cm-3)
C                            1st DIMENSION: ELECTRON DENSITY     INDEX
C                            2nd DIMENSION: DATA-BLOCK INDEX
C          (R*8)   PEC(,,)   =INPUT DATA SET -
C                             FULL SET OF IONIZATIONS PER PHOTON
C                             1st DIMENSION: ELECTRON TEMPERATURE INDEX
C                             2nd DIMENSION: ELECTRON DENSITY     INDEX
C                             3rd DIMENSION: DATA-BLOCK INDEX
C
C          (C*10)  CWAVEL() = INPUT FILE - WAVELENGTH (ANGSTROMS)
C                             DIMENSION: DATA-BLOCK INDEX
C          (C*8)   CFILE()  = INPUT FILE - SPECIFIC ION FILE SOURCE
C                             DIMENSION: DATA-BLOCK INDEX
C          (C*8)   CTYPE()  = INPUT FILE - TYPE OF DATA (IE EXCIT., ETC)
C                             DIMENSION: DATA-BLOCK INDEX
C          (C*2)   CINDM()  = INPUT FILE - METASTABLE INDEX
C                             DIMENSION: DATA-BLOCK INDEX
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          E3FILE     ADAS      OPEN DATA SET FOR SELECTED EMITTER
C          E3DATA     ADAS      FETCH INPUT DATA FROM SELECTED DATA SET
C          E3CHKB     ADAS      CHECK VALIDITY OF ION AND 'IBSEL'
C          E3SPLN     ADAS      INTERPOLATE DATA WITH TWO WAY SPLINES
C          E3TITL     ADAS      CREATE DESCRIPTIVE TITLE FOR OUTPUT
C          XXUID      ADAS      FETCHES/SETS ADAS SOURCE DATA USER ID
C          XXSPEC     ADAS      FETCHES/SETS ADAS SOURCE DATA FILE NAME+
C
C AUTHOR:  H. P. SUMMERS
C          K1/1/57
C          JET EXT. 4941
C
C DATE:    11/10/91
C
C UPDATE:  05/12/91 - PE BRIDEN: 'NSTORE' INCREASED FROM 10 TO 100
C
C UPDATE:  28/02/92 - PE BRIDEN: 'NSTORE' INCREASED FROM 100 TO 150
C
C UPDATE:  10/03/93 - PE BRIDEN: INTRODUCED CALL TO XXUID TO ESTABLISH
C                                IF USERID OF INPUT DATASET CHANGES
C                                BETWEEN CALLS.
C                                SAVE NAME OF LAST READ DATASET.
C                                (ADDED VARIABLES UIDIN,UIDLST,DSNREQ)
C
C UPDATE:   2/09/93 - HPS      : INTRODUCED CALL TO XXSPEC TO ESTABLISH
C                                IF USRGRP, USRTYP AND USREXT OF INPUT
C                                DATASET CHANGES BETWEEN CALLS.
```

```
C                                SAVE NAME OF LAST READ DATASET.
C                                (ADDED VARIABLES GRPIN,GRPLST,TYPIN,
C                                 TYPLST, EXTIN, EXTLST)
C
C UPDATE:   6/05/94 - PEB      : INCREASED PARAMETER NSTORE 150 -> 350
C
C UPDATE:   3/11/94 - L.JALOTA : CHANGED DSNAME, UIDIN SIZE TO 80 CHARS.
C UPDATE:  23/11/94 - L.JALOTA : TIDIED UP STRING LENGTH DEFINITIONS
C
C VERSION: 1.1                 DATE: 25-05-95
C MODIFIED: TIM HAMMOND
C           - PUT UNDER S.C.C.S. CONTROL
C
C VERSION: 1.2                 DATE: 22-04-96
C MODIFIED: TIM HAMMOND/PAUL BRIDEN
C           - INCREASED PARAMETERS: NSTORE: 350 -> 500
C                                   NDDIM:   20 -> 26
C                                   NTDIM:   20 -> 35
C           - MODIFIED CONVERSION OF CWAVEL->WLNGTH
C             (ADDED VARIABLES C11 AND IPOS)
C
C VERSION: 1.3 DATE: 20-09-99
C MODIFIED: RICHARD MARTIN
C INCREASED TITLX TO CHAR*120
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      CHARACTER*120       TITLX
      INTEGER             IBSEL,       IRCODE,      ITVAL,       IZ0IN
      INTEGER             IZIN
      LOGICAL             LDRNG(ITVAL),             LTRNG(ITVAL)
      REAL*8              DVAL(ITVAL), PECA(ITVAL), TVAL(ITVAL), WLNGTH
```

6.40 spzd: Subroutine spzd from library adas5xx

```
      SUBROUTINE SPZD( IBSEL  , IZ0IN  ,
     &                 ITVAL  , TVAL   ,
     &                 IZ     , IZ1  ,
     &                 PZDA   , LTRNG  ,
     &                 TITLX  , IRCODE
     &               )
C-----------------------------------------------------------------------
C
C  ******************* FORTRAN77 SUBROUTINE: SPZD *********************
C
C  PURPOSE: TO EXTRACT AND INTERPOLATE  DENSITY INDEPENDENT  RADIATED
C           POWER COEFFICIENTS FOR GIVEN ELEMENT NUCLEAR CHARGE AND
C           DATA-BLOCK FOR A SET OF ELECTRON TEMPERATURES.
C
C           USES THE SAME ROUTINES AS ADAS504, EXCEPT FOR:
C
C           'E4FILE' - WHICH OPENS THE REQUESTED FILE.
C           'E4CHKB' - WHICH CHECKS INPUT  VALUES  ARE  CONSISTENT  WITH
C                      THE SELECTED DATA-BLOCK 'IBSEL' AND   'IBSEL'  IS
C                      IN RANGE.
C
C            THE FIRST OF THESE FUNCTIONS IS CARRIED  OUT  IN  'ADAS604'
C            VIA ISPF PANELS USING THE ROUTINE 'E4SPF0'  -  ADAS604 DOES
C            NOT REQUIRE THE ROUTINE 'E4CHKB' AS THE USER CANNOT  SELECT
C            AN INVALID VALUE FOR 'IBSEL' OR 'IBSEL'/ELEMENT COMBINATION
C
C
C  CALLING PROGRAM: GENERAL USE
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  IBSEL   = INDEX OF DATA-BLOCK SELECTED FOR ANALYSIS
C  INPUT : (I*4)  IZ0IN   = NUCLEAR CHARGE OF REQUIRED ELEMENT
C
C  INPUT : (I*4)  ITVAL   = NUMBER OF ELECTRON TEMPERATURE VALUES
C  INPUT : (R*8)  TVAL()  = ELECTRON TEMPERATUIRES (UNITS: EV)
C                           DIMENSION: ELECTRON TEMPERATURE INDEX
C
C  OUTPUT: (I*4)  IZ      = INPUT FILE - SELECTED DATA BLOCK:
C                           RADIATING ION CHARGE
C                                (SET TO -1 WHEN WHOLE ELEMENT)
C  OUTPUT: (I*4)  IZ1     = INPUT FILE - SELECTED DATA BLOCK:
C                           RADIATING ION CHARGE +1
C                                ( SET TO 1 WHEN WHOLE ELEMENT)
C
C  OUTPUT: (R*8)  PZDA()  = ZERO-DENSITY RADIATED POWER COEFFICIENTS
C                           DIMENSION: ELECTRON TEMPERATURE INDEX
C  OUTPUT: (L*4)  LTRNG() =.TRUE.  => OUTPUT 'PZDA()'  VALUE WAS INTER-
C                                     POLATED  FOR  THE  USER  ENTERED
C                                     ELECTRON TEMPERATURE 'TVAL()'.
C                          .FALSE. => OUTPUT 'PZDA()'  VALUE WAS EXTRA-
C                                     POLATED  FOR  THE  USER  ENTERED
C                                     ELECTRON TEMPERATURE 'TVAL()'.
C                           DIMENSION: ELECTRON TEMPERATURE INDEX
C
C  OUTPUT: (C*80) TITLX   = INFORMATION STRING (DSN ETC.)
C  OUTPUT: (I*4)  IRCODE  = RETURN CODE FROM SUBROUTINE:
C                           0 => NORMAL COMPLETION - NO ERROR DETECTED
C                           1 => DATA SET MEMBER FOR IONIZING ION WITH
```

```
C                               NUCLEAR CHARGE  'IZ0IN'  CAN  NOT  BE
C                               FOUND/DOES NOT EXIST.
C                             2 => DISCREPANCY BETWEEN REQUESTED CHARGES
C                               AND THOSE IN INPUT FILE.
C                             3 => THE SELECTED DATA-BLOCK 'IBSEL' IS OUT
C                               OF RANGE OR DOES NOT EXIST.
C                             4 => INVALID VALUE FOR 'IZ0IN' ENTERED.
C                               ('IZ0MIN' <= 'IZ0IN' <= 'IZ0MAX')
C                             9 => ERROR ENCOUNTERED WHEN TRYING TO OPEN
C                               INPUT DATA-SET.
C
C          (I*4)  NSTORE  = PARAMETER= MAXIMUM NUMBER  OF  DATA-BLOCKS
C                                      WHICH CAN BE READ FROM THE INPUT
C                                      DATA-SET.
C          (I*4)  NTDIM   = PARAMETER= MAXIMUM NUMBER OF ELECTRON TEMP-
C                                      ERATURES THAT CAN BE READ  FROM
C                                      AN INPUT DATA-SET DATA-BLOCK.
C          (I*4)  IZ0MIN  = PARAMETER: MIN. ALLOWED VALUE FOR 'IZ0IN'
C          (I*4)  IZ0MAX  = PARAMETER: MAX. ALLOWED VALUE FOR 'IZ0IN'
C
C          (I*4)  IZ0LST  = LAST VALUE OF 'IZ0IN' FOR  WHICH  INPUT
C                           DATA WAS READ.
C          (I*4)  IUNIT   = UNIT TO WHICH INPUT DATA SET IS ALLOCATED
C          (I*4)  NBSEL   = TOTAL NUMBER OF DATA-BLOCKS READ FROM INPUT
C                           DATA SET.
C          (I*4)  IZ0     = INPUT FILE - EMITTING ION - NUCLEAR CHARGE
C
C          (L*4)  LOPEN   = .TRUE.  => INPUT DATA SET OPEN.
C                           .FALSE. => INPUT DATA SET CLOSED.
C
C          (C*2)  ESYM    = INPUT FILE - IONIZING ION - ELEMENT SYMBOL
C          (C*3)  EXTIN   = CURRENT ADAS SOURCE DATA FILE EXTENSION
C          (C*3)  EXTLST  = ADAS SOURCE DATA FILE EXT. USED LAST TIME
C                           DATA WAS READ.
C          (C*80) UIDIN   = CURRENT ADAS SOURCE DATA USER ID.
C          (C*80) UIDLST  = ADAS SOURCE DATA USER ID USED LAST TIME
C                           DATA WAS READ.
C          (C*8)  GRPIN   = CURRENT ADAS SOURCE DATA GROUPNAME
C          (C*8)  GRPLST  = ADAS SOURCE DATA GROUPNAME USED LAST TIME
C                           DATA WAS READ.
CA     (C*80) TYPIN   = CURRENT SUB-DIRECTORY (OPTIONAL)
CA         (C*80) TYPLST  = SUB-DIRECTORY USED LAST TIME DATA WAS READ
C
C
C          (C*80) DSNREQ  = NAME OF DATA SET REQUESTED
C                           (MAY OR MAY NOT EXIST)
C          (C*80) DSNAME  = FULL NAME OF DATA SET INTERROGATED
C
C          (I*4)  ISELA() = INPUT DATA FILE: DATA-BLOCK ENTRY INDICES.
C                           DIMENSION: DATA-BLOCK INDEX
C          (I*4)  ITA()   = INPUT DATA SET-NUMBER OF ELECTRON TEMPERA-
C                           TURES.
C                           DIMENSION: DATA-BLOCK INDEX
C          (I*4)  IZOUT() = INPUT DATA FILE: IONIZING ION INITIAL CHARGE
C                           DIMENSION: DATA-BLOCK INDEX
C          (I*4)  IZ1OUT()= INPUT DATA FILE: IONIZING ION FINAL   CHARGE
C                           DIMENSION: DATA-BLOCK INDEX
C
C          (R*8)  TETA(,) = INPUT DATA SET -
C                           ELECTRON TEMPERATURES (UNITS: eV)
C                           1st DIMENSION: ELECTRON TEMPERATURE INDEX
```

```
C                           2nd DIMENSION: DATA-BLOCK INDEX
C          (R*8)  PZD(,)    =INPUT DATA SET -
C                            FULL SET OF IONIZATIONS RATE-COEFFICIENTS
C                            1st DIMENSION: ELECTRON TEMPERATURE INDEX
C                            2nd DIMENSION: DATA-BLOCK INDEX
C
C          (C*5)  CIION() = INPUT DATA FILE - RADIATING ION
C                           DIMENSION: DATA-BLOCK INDEX
C          (C*5)  CITYPE()= INPUT DATA FILE - RADIATION TYPE
C                           DIMENSION: DATA-BLOCK INDEX
C          (C*20) CIINFO()= INPUT DATA FILE - INFORMATION STRING
C                           DIMENSION: DATA-BLOCK INDEX
C
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          E4FILE     ADAS      OPEN DATA SET FOR SELECTED ELEMENT
C          E4DATA     ADAS      FETCH INPUT DATA FROM SELECTED DATA SET
C          E4CHKB     ADAS      CHECK VALIDITY OF ELEMENT AND 'IBSEL'
C          E4SPLN     ADAS      INTERPOLATE DATA WITH ONE-WAY SPLINES
C          E4TITL     ADAS      CREATE DESCRIPTIVE TITLE FOR OUTPUT
C          XXUID      ADAS      FETCHES/SETS ADAS SOURCE DATA USER ID
C          XXSPZD     ADAS      FETCHES/SETS ADAS SOURCE DATA FILE NAME+
C
C AUTHOR:  H.P. SUMMERS
C          K1/1/47
C          JET EXT. 4941
C
C DATE:    28/08/92
C
C UPDATE:  10/03/93 - PE BRIDEN: INTRODUCED CALL TO XXUID TO ESTABLISH
C                                IF USERID OF INPUT DATASET CHANGES
C                                BETWEEN CALLS.
C                                SAVE NAME OF LAST READ DATASET.
C                                (ADDED VARIABLES UIDIN,UIDLST,DSNREQ)
C
C UPDATE:   2/09/93 - HPS      : INTRODUCED CALL TO XXSPZD TO ESTABLISH
C                                IF USRGRP, USRTYP AND USREXT OF INPUT
C                                DATASET CHANGES BETWEEN CALLS.
C                                SAVE NAME OF LAST READ DATASET.
C                                (ADDED VARIABLES GRPIN,GRPLST,TYPIN,
C                                 TYPLST, EXTIN, EXTLST)
C
C UPDATE:   4/11/94 - L. JALOTA: MODIFIED TO RUN UNDER UNIX.
C   CHANGED DIMENSIONS OF DSNAME,DSNREQ
C
C UPDATE:  26/03/95 - HPS      : CORRECTED CIION and CITYP TO C*5
C-----------------------------------------------------------------------
C
C VERSION: 1.1 DATE: 25-05-95
C MODIFIED: UNKNOWN (SOMEONE FROM TESSELLA SUPPORT SERVICES PLC)
C     - PUT UNDER SCCS CONTROL
C
C VERSION: 1.2 DATE: 20-07-07
C MODIFIED: Allan Whiteford
C     - Small modification to comments to allow for automatic
C             documentation preparation.
C
C-----------------------------------------------------------------------
C
```

```
C-----------------------------------------------------------------------
      CHARACTER*80        TITLX
      INTEGER             IBSEL,       IRCODE,      ITVAL,       IZ
      INTEGER             IZ0IN,       IZ1
      LOGICAL             LTRNG(ITVAL)
      REAL*8              PZDA(ITVAL), TVAL(ITVAL)
```

6.41 ssxb: Subroutine ssxb from library adas5xx

```
      SUBROUTINE SSXB( IBSEL  , IZIN   , IZ0IN  ,
     &                 ITVAL  , TVAL   , DVAL   ,
     &                 WLNGTH ,
     &                 SXBA   , LTRNG  , LDRNG  ,
     &                 TITLX  , IRCODE
     &               )
C-----------------------------------------------------------------------
C
C  ******************* FORTRAN77 SUBROUTINE: SSXB *********************
C
C  PURPOSE: TO EXTRACT AND INTERPOLATE  IONIZATIONS PER PHOTON FOR
C           EMITTING IONS.
C           USES THE SAME ROUTINES USED BY ADAS501, EXCEPT FOR:
C
C           'E1FILE' - WHICH OPENS THE REQUESTED FILE.
C           'E1CHKB' - WHICH CHECKS INPUT  VALUES  ARE  CONSISTENT  WITH
C                      THE SELECTED DATA-BLOCK 'IBSEL' AND   'IBSEL'  IS
C                      IN RANGE.
C
C            THE FIRST OF THESE FUNCTIONS IS CARRIED  OUT  IN  'ADAS501'
C            VIA ISPF PANELS USING THE ROUTINE 'E1SPF0'  -  ADAS501 DOES
C            NOT REQUIRE THE ROUTINE 'E1CHKB' AS THE USER CANNOT  SELECT
C            AN INVALID VALUE FOR 'IBSEL' OR 'IBSEL'/EMITTER COMBINATION
C
C
C  CALLING PROGRAM: GENERAL USE
C
C  SUBROUTINE:
C
C  INPUT : (I*4)   IBSEL   = INDEX OF DATA-BLOCK SELECTED FOR ANALYSIS
C  INPUT : (I*4)   IZIN    = ION CHARGE OF EMITTING ION
C  INPUT : (I*4)   IZ0IN   = NUCLEAR CHARGE OF EMITTING ION
C
C  INPUT : (I*4)   ITVAL   = NO. OF ELECTRON TEMPERATURE/DENSIY PAIRS
C  INPUT : (R*8)   TVAL()  = ELECTRON TEMPERATUIRES (UNITS: EV)
C                            DIMENSION: TEMPERATURE/DENSITY PAIR INDEX
C  INPUT : (R*8)   DVAL()  = ELECTRON DENSITIES (UNITS: CM-3)
C                            DIMENSION: TEMPERATURE/DENSITY PAIR INDEX
C
C  OUTPUT: (R*8)   WLNGTH  = SELECTED BLOCK WAVELENGTH (ANGSTROMS)
C
C  OUTPUT: (R*8)   SXBA()  = IONIZATIONS PER PHOTON.
C                            DIMENSION: TEMPERATURE/DENSITY PAIR INDEX
C  OUTPUT: (L*4)   LTRNG() =.TRUE.  => OUTPUT 'SXBA()'  VALUE WAS INTER-
C                                      POLATED  FOR  THE  USER  ENTERED
C                                      ELECTRON TEMPERATURE 'TVAL()'.
C                           .FALSE. => OUTPUT 'SXBA()'  VALUE WAS EXTRA-
C                                      POLATED  FOR  THE  USER  ENTERED
C                                      ELECTRON TEMPERATURE 'TVAL()'.
C                            DIMENSION: TEMPERATURE/DENSITY PAIR INDEX
C  OUTPUT: (L*4)   LDRNG() =.TRUE.  => OUTPUT 'SXBA()'  VALUE WAS INTER-
C                                      POLATED  FOR  THE  USER  ENTERED
C                                      ELECTRON DENSITY 'DVAL()'.
C                           .FALSE. => OUTPUT 'SXBA()'  VALUE WAS EXTRA-
C                                      POLATED  FOR  THE  USER  ENTERED
C                                      ELECTRON DENSITY 'DVAL()'.
C                            DIMENSION: TEMPERATURE/DENSITY PAIR INDEX
C
C  OUTPUT: (C*80)  TITLX   = INFORMATION STRING (DSN ETC.)
```

```
C  OUTPUT: (I*4)   IRCODE  = RETURN CODE FROM SUBROUTINE:
C                            0 => NORMAL COMPLETION - NO ERROR DETECTED
C                            1 => DATA SET MEMBER FOR EMITTING ION WITH
C                                 CHARGE 'IZIN' & ION CHARGE 'IZ0IN' CAN
C                                 NOT BE FOUND/DOES NOT EXIST.
C                            2 => DISCREPANCY BETWEEN REQUESTED CHARGES
C                                 AND THOSE IN INPUT FILE.
C                            3 => THE SELECTED DATA-BLOCK 'IBSEL' IS OUT
C                                 OF RANGE OR DOES NOT EXIST.
C                            4 => INVALID VALUE FOR 'IZ0IN' ENTERED.
C                                 ('IZ0MIN' <= 'IZ0IN' <= 'IZ0MAX')
C                            5 => INVALID VALUE FOR 'IZIN' ENTERED.
C                                 ( 0  <= 'IZIN' <= 99 )
C                            9 => ERROR ENCOUNTERED WHEN TRYING TO OPEN
C                                 INPUT DATA-SET.
C
C          (I*4)   NSTORE  = PARAMETER= MAXIMUM NUMBER  OF  DATA-BLOCKS
C                                       WHICH CAN BE READ FROM THE INPUT
C                                       DATA-SET.
C          (I*4)   NTDIM   = PARAMETER= MAXIMUM NUMBER OF ELECTRON TEMP-
C                                       ERATURES THAT CAN BE READ  FROM
C                                       AN INPUT DATA-SET DATA-BLOCK.
C          (I*4)   NDDIM   = PARAMETER= MAXIMUM NUMBER OF ELECTRON DENS-
C                                       ITIES  THAT  CAN  BE  READ  FROM
C                                       AN INPUT DATA-SET DATA-BLOCK.
C          (I*4)   IZ0MIN  = PARAMETER: MIN. ALLOWED VALUE FOR 'IZ0IN'
C          (I*4)   IZ0MAX  = PARAMETER: MAX. ALLOWED VALUE FOR 'IZ0IN'
C
C          (I*4)   IZ0LST  = LAST VALUE OF 'IZ0IN' FOR  WHICH  INPUT
C                            DATA WAS READ.
C          (I*4)   IZLAST  = LAST VALUE OF 'IZIN' FOR  WHICH  INPUT
C                            DATA WAS READ.
C          (I*4)   IUNIT   = UNIT TO WHICH INPUT DATA SET IS ALLOCATED
C          (I*4)   NBSEL   = TOTAL NUMBER OF DATA-BLOCKS READ FROM INPUT
C                            DATA SET.
C          (I*4)   IZ0     = INPUT FILE - EMITTING ION - NUCLEAR CHARGE
C          (I*4)   IZ      = INPUT FILE - EMITTING ION - CHARGE
C          (I*4)   IZ1     = INPUT FILE - EMITTING ION - CHARGE + 1
C          (I*4)   IPOS    = USED IN CONVERTING CWAVEL -> WLNGTH
C
C          (L*4)   LOPEN   = .TRUE.  => INPUT DATA SET OPEN.
C                            .FALSE. => INPUT DATA SET CLOSED.
C
C          (C*2)   ESYM    = INPUT FILE - EMITTING ION - ELEMENT SYMBOL
C          (C*3)   EXTIN   = CURRENT ADAS SOURCE DATA FILE EXTENSION
C          (C*3)   EXTLST  = ADAS SOURCE DATA FILE EXT. USED LAST TIME
C                            DATA WAS READ.
CA         (C*80)  UIDIN   = CURRENT ADAS SOURCE DATA PATH
CA    (C*80)  UIDLST  = ADAS SOURCE DATA PATH USED LAST TIME
C                            DATA WAS READ.
C          (C*8)   GRPIN   = CURRENT ADAS SOURCE DATA GROUPNAME
C          (C*8)   GRPLST  = ADAS SOURCE DATA GROUPNAME USED LAST TIME
C                            DATA WAS READ.
CA         (C*80)  TYPIN   = OPTIONAL SUBDIRECTORY FOR FILE
CA         (C*80)  TYPLST  = OPTIONAL SUBDIRECTORY FOR FILE USED LAST TIME
C                            DATA WAS READ.
C          (C*11)  C11     = USED IN CONVERTING CWAVEL -> WLNGTH
C          (C*80)  DSNREQ  = NAME OF DATA FILE REQUESTED INCLUDING PATH
C                            (MAY OR MAY NOT EXIST)
C          (C*80)  DSNAME  = FULL NAME OF DATA FILE INTERROGATED
C
```

```
C          (I*4)   ISELA() = INPUT DATA FILE: DATA-BLOCK ENTRY INDICES.
C                            DIMENSION: DATA-BLOCK INDEX
C          (I*4)   ITA()   = INPUT DATA SET-NUMBER OF ELECTRON TEMPERA-
C                            TURES.
C                            DIMENSION: DATA-BLOCK INDEX
C          (I*4)   IDA()   = INPUT DATA SET-NUMBER OF ELECTRON DENSITIES
C                            DIMENSION: DATA-BLOCK INDEX
C
C          (R*8)   TETA(,) = INPUT DATA SET -
C                            ELECTRON TEMPERATURES (UNITS: eV)
C                            1st DIMENSION: ELECTRON TEMPERATURE INDEX
C                            2nd DIMENSION: DATA-BLOCK INDEX
C          (R*8)   TEDA(,) = INPUT DATA SET -
C                            ELECTRON DENSITIES    (UNITS: cm-3)
C                            1st DIMENSION: ELECTRON DENSITY     INDEX
C                            2nd DIMENSION: DATA-BLOCK INDEX
C          (R*8)   SXB(,,)   =INPUT DATA SET -
C                             FULL SET OF IONIZATIONS PER PHOTON
C                             1st DIMENSION: ELECTRON TEMPERATURE INDEX
C                             2nd DIMENSION: ELECTRON DENSITY     INDEX
C                             3rd DIMENSION: DATA-BLOCK INDEX
C
C          (C*10)  CWAVEL() = INPUT FILE - WAVELENGTH (ANGSTROMS)
C                             DIMENSION: DATA-BLOCK INDEX
C          (C*8)   CFILE()  = INPUT FILE - SPECIFIC ION FILE SOURCE
C                             DIMENSION: DATA-BLOCK INDEX
C          (C*8)   CPCODE() = INPUT FILE - SPECIFIC ION PROCESSING CODE
C                             DIMENSION: DATA-BLOCK INDEX
C          (C*2)   CINDM()  = INPUT FILE - METASTABLE INDEX
C                             DIMENSION: DATA-BLOCK INDEX
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          E1FILE     ADAS      OPEN DATA SET FOR SELECTED EMITTER
C          E1DATA     ADAS      FETCH INPUT DATA FROM SELECTED DATA SET
C          E1CHKB     ADAS      CHECK VALIDITY OF ION AND 'IBSEL'
C          E1SPLN     ADAS      INTERPOLATE DATA WITH TWO WAY SPLINES
C          E1TITL     ADAS      CREATE DESCRIPTIVE TITLE FOR OUTPUT
C          XXUID      ADAS      FETCHES/SETS ADAS SOURCE DATA USER ID
C          XXSSXB     ADAS      FETCHES/SETS ADAS SOURCE DATA FILE NAME+
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/37
C          JET EXT. 5023
C
C DATE:    30/04/91
C
C UPDATE:  05/12/91 - PE BRIDEN: 'NSTORE' INCREASED FROM 10 TO 100
C
C UPDATE:  28/02/92 - PE BRIDEN: 'NSTORE' INCREASED FROM 100 TO 150
C
C UPDATE:  10/03/93 - PE BRIDEN: INTRODUCED CALL TO XXUID TO ESTABLISH
C                                IF USERID OF INPUT DATASET CHANGES
C                                BETWEEN CALLS.
C                                SAVE NAME OF LAST READ DATASET.
C                                (ADDED VARIABLES UIDIN,UIDLST,DSNREQ)
C
C UPDATE:   2/09/93 - HPS      : INTRODUCED CALL TO XXSSXB TO ESTABLISH
C                                IF USRGRP, USRTYP AND USREXT OF INPUT
C                                DATA SET CHANGES BETWEEN CALLS.
```

```
C                                SAVE NAME OF LAST READ DATASET.
C                                (ADDED VARIABLES GRPIN,GRPLST,TYPIN,
C                                 TYPLST, EXTIN, EXTLST)
C UPDATE:   31/10/03 - LALIT JALOTA
C CHANGED STRING DIMENSIONS TO 80
C
C VERSION: 1.1                          DATE: 25-05-95
C MODIFIED:  TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - PUT UNDER S.C.C.S. CONTROL
C
C VERSION: 1.2                          DATE: 22-04-96
C MODIFIED:  TIM HAMMOND
C               - INCREASED PARAMETERS: NSTORE 150 -> 250
C                                       NTDIM   20 ->  35
c                                       NDDIM   20 ->  26
C               - MODIFIED CONVERSION OF CWAVEL->WLNGTH
C                 (ADDED VARIABLES C11 AND IPOS)
c
C VERSION: 1.3                          DATE: 17-01-2001
C MODIFIED:  Martin O'Mullane
C               - Removed spurious 2 in 1st column of a comment.
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      CHARACTER*80        TITLX
      INTEGER             IBSEL,       IRCODE,      ITVAL,       IZ0IN
      INTEGER             IZIN
      LOGICAL             LDRNG(ITVAL),             LTRNG(ITVAL)
      REAL*8              DVAL(ITVAL), SXBA(ITVAL), TVAL(ITVAL), WLNGTH
```

6.42 sszd: Subroutine sszd from library adas5xx

```
      SUBROUTINE SSZD( IBSEL  , IZ0IN  ,
     &                 ITVAL  , TVAL   ,
     &                 BWNO   , IZ     , IZ1  ,
     &                 METI   , METF   ,
     &                 SZDA   , LTRNG  ,
     &                 TITLX  , IRCODE
     &               )
C-----------------------------------------------------------------------
C
C  ******************* FORTRAN77 SUBROUTINE: SSZD *********************
C
C  PURPOSE: TO EXTRACT AND  INTERPOLATE  ZERO-DENSITY  IONIZATION  RATE-
C           COEFFICIENTS FOR GIVEN ELEMENT NUCLEAR CHARGE AND DATA-BLOCK
C           FOR AN INPUT SET OF ELECTRON TEMPERATURES.
C
C           USES THE SAME ROUTINES USED BY ADAS502, EXCEPT FOR:
C
C           'E2FILE' - WHICH OPENS THE REQUESTED FILE.
C           'E2CHKB' - WHICH CHECKS INPUT  VALUES  ARE  CONSISTENT  WITH
C                      THE SELECTED DATA-BLOCK 'IBSEL' AND   'IBSEL'  IS
C                      IN RANGE.
C
C            THE FIRST OF THESE FUNCTIONS IS CARRIED  OUT  IN  'ADAS502'
C            VIA ISPF PANELS USING THE ROUTINE 'E2SPF0'  -  ADAS502 DOES
C            NOT REQUIRE THE ROUTINE 'E2CHKB' AS THE USER CANNOT  SELECT
C            AN INVALID VALUE FOR 'IBSEL' OR 'IBSEL'/ELEMENT COMBINATION
C
C
C  CALLING PROGRAM: GENERAL USE
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  IBSEL   = INDEX OF DATA-BLOCK SELECTED FOR ANALYSIS
C  INPUT : (I*4)  IZ0IN   = NUCLEAR CHARGE OF REQUIRED ELEMENT
C
C  INPUT : (I*4)  ITVAL   = NUMBER OF ELECTRON TEMPERATURE VALUES
C  INPUT : (R*8)  TVAL()  = ELECTRON TEMPERATUIRES (UNITS: EV)
C                           DIMENSION: ELECTRON TEMPERATURE INDEX
C
C  OUTPUT: (R*8)  BWNO    = INPUT FILE - SELECTED DATA-BLOCK:
C                           EFFECTIVE IONIZATION POTENTIAL (cm-1).
C  OUTPUT: (I*4)  IZ      = INPUT FILE - SELECTED DATA BLOCK:
C                           IONIZING ION - INITIAL CHARGE
C  OUTPUT: (I*4)  IZ1     = INPUT FILE - SELECTED DATA BLOCK:
C                           IONIZING ION - FINAL   CHARGE
C
C  OUTPUT: (I*4)  METI    = INPUT FILE - SELECTED DATA-BLOCK:
C                           INITIAL STATE METSTABLE INDEX
C  OUTPUT: (I*4)  METF    = INPUT FILE - SELECTED DATA-BLOCK:
C                           FINAL   STATE METSTABLE INDEX
C
C  OUTPUT: (R*8)  SZDA()  = ZERO-DENSITY IONIZATION RATE-COEFFICIENTS
C                           DIMENSION: ELECTRON TEMPERATURE INDEX
C  OUTPUT: (L*4)  LTRNG() =.TRUE.  => OUTPUT 'SZDA()'  VALUE WAS INTER-
C                                     POLATED  FOR  THE  USER  ENTERED
C                                     ELECTRON TEMPERATURE 'TVAL()'.
C                          .FALSE. => OUTPUT 'SZDA()'  VALUE WAS EXTRA-
C                                     POLATED  FOR  THE  USER  ENTERED
C                                     ELECTRON TEMPERATURE 'TVAL()'.
```

```
C                           DIMENSION: ELECTRON TEMPERATURE INDEX
C
C  OUTPUT: (C*80) TITLX   = INFORMATION STRING (DSN ETC.)
C  OUTPUT: (I*4)  IRCODE  = RETURN CODE FROM SUBROUTINE:
C                           0 => NORMAL COMPLETION - NO ERROR DETECTED
C                           1 => DATA SET MEMBER FOR IONIZING ION WITH
C                                NUCLEAR CHARGE 'IZ0IN' CAN NOT BE
C                                FOUND/DOES NOT EXIST.
C                           2 => DISCREPANCY BETWEEN REQUESTED CHARGES
C                                AND THOSE IN INPUT FILE.
C                           3 => THE SELECTED DATA-BLOCK 'IBSEL' IS OUT
C                                OF RANGE OR DOES NOT EXIST.
C                           4 => INVALID VALUE FOR 'IZ0IN' ENTERED.
C                                ('IZ0MIN' <= 'IZ0IN' <= 'IZ0MAX')
C                           9 => ERROR ENCOUNTERED WHEN TRYING TO OPEN
C                                INPUT DATA-SET.
C
C          (I*4)  NSTORE  = PARAMETER= MAXIMUM NUMBER OF DATA-BLOCKS
C                                      WHICH CAN BE READ FROM THE INPUT
C                                      DATA-SET.
C          (I*4)  NTDIM   = PARAMETER= MAXIMUM NUMBER OF ELECTRON TEMP-
C                                      ERATURES THAT CAN BE READ FROM
C                                      AN INPUT DATA-SET DATA-BLOCK.
C          (I*4)  IZ0MIN  = PARAMETER: MIN. ALLOWED VALUE FOR 'IZ0IN'
C          (I*4)  IZ0MAX  = PARAMETER: MAX. ALLOWED VALUE FOR 'IZ0IN'
C
C          (I*4)  IZ0LST  = LAST VALUE OF 'IZ0IN' FOR WHICH INPUT
C                           DATA WAS READ.
C          (I*4)  IUNIT   = UNIT TO WHICH INPUT DATA SET IS ALLOCATED
C          (I*4)  NBSEL   = TOTAL NUMBER OF DATA-BLOCKS READ FROM INPUT
C                           DATA SET.
C          (I*4)  IZ0     = INPUT FILE - EMITTING ION - NUCLEAR CHARGE
C
C          (L*4)  LOPEN   = .TRUE.  => INPUT DATA SET OPEN.
C                           .FALSE. => INPUT DATA SET CLOSED.
C
C          (C*2)  ESYM    = INPUT FILE - IONIZING ION - ELEMENT SYMBOL
C          (C*3)  EXTIN   = CURRENT ADAS SOURCE DATA FILE EXTENSION
C          (C*3)  EXTLST  = ADAS SOURCE DATA FILE EXT. USED LAST TIME
C                           DATA WAS READ.
C          (C*6)  UIDIN   = CURRENT ADAS SOURCE DATA USER ID.
CA         (C*80) UIDIN   = CURRENT ADAS SOURCE DATA FILE PATH
CA         (C*80) UIDLST  = ADAS SOURCE DATA FILE PATH USED LAST TIME
C                           DATA WAS READ.
C          (C*8)  GRPIN   = CURRENT ADAS SOURCE DATA GROUPNAME
C          (C*8)  GRPLST  = ADAS SOURCE DATA GROUPNAME USED LAST TIME
C                           DATA WAS READ.
CA         (C*80) TYPIN   = CURRENT ADAS FILE SUBDIRECTORY( OPTIONAL)
CA         (C*80) TYPLST  = ADAS FILE SUBDIRECTORY USED LAST TIME (OPT)
C                           DATA WAS READ.
CA    (C*80) DSNREQ  = DATAFILE NAME UNDER UNIX INCLUDING PATH
C                           (MAY OR MAY NOT EXIST)
CA         (C*80) DSNAME  = DATAFILE NAME UNDER UNIX INCLUDING PATH
C
C          (I*4)  ISELA() = INPUT DATA FILE: DATA-BLOCK ENTRY INDICES.
C                           DIMENSION: DATA-BLOCK INDEX
C          (I*4)  ITA()   = INPUT DATA SET-NUMBER OF ELECTRON TEMPERA-
C                           TURES.
C                           DIMENSION: DATA-BLOCK INDEX
C          (I*4)  IZOUT() = INPUT DATA FILE: IONIZING ION INITIAL CHARGE
C                           DIMENSION: DATA-BLOCK INDEX
```

```
C          (I*4)  IZ1OUT()= INPUT DATA FILE: IONIZING ION FINAL   CHARGE
C                          DIMENSION: DATA-BLOCK INDEX
C
C          (R*8)  BWNOUT()= INPUT DATA FILE: EFFECTIVE IONIZATION POT.
C                          (UNITS: cm-1).
C                          DIMENSION: DATA-BLOCK INDEX
C          (R*8)  TETA(,) = INPUT DATA SET -
C                          ELECTRON TEMPERATURES (UNITS: eV)
C                          1st DIMENSION: ELECTRON TEMPERATURE INDEX
C                          2nd DIMENSION: DATA-BLOCK INDEX
C          (R*8)  SZD(,)    =INPUT DATA SET -
C                           FULL SET OF IONIZATIONS RATE-COEFFICIENTS
C                           1st DIMENSION: ELECTRON TEMPERATURE INDEX
C                           3rd DIMENSION: DATA-BLOCK INDEX
C
C          (C*2)  CICODE()= INPUT DATA FILE - INITIAL STATE META. INDEX
C                          DIMENSION: DATA-BLOCK INDEX
C          (C*2)  CFCODE()= INPUT DATA FILE - FINAL   STATE META. INDEX
C                          DIMENSION: DATA-BLOCK INDEX
C          (C*6)  CIION() = INPUT DATA FILE - INITIAL ION
C                          DIMENSION: DATA-BLOCK INDEX
C          (C*6)  CFION() = INPUT DATA FILE - FINAL   ION
C                          DIMENSION: DATA-BLOCK INDEX
C
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          E2FILE     ADAS      OPEN DATA SET FOR SELECTED ELEMENT
C          XXDATA_07  ADAS      FETCH INPUT DATA FROM SELECTED DATA SET
C          E2CHKB     ADAS      CHECK VALIDITY OF ELEMENT AND 'IBSEL'
C          E2SPLN     ADAS      INTERPOLATE DATA WITH ONE-WAY SPLINES
C          E2TITL     ADAS      CREATE DESCRIPTIVE TITLE FOR OUTPUT
C          XXUID      ADAS      FETCHES/SETS ADAS SOURCE DATA USER ID
C          XXSPEC     ADAS      FETCHES/SETS ADAS SOURCE DATA FILE NAME+
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/37
C          JET EXT. 6023
C
C DATE:    07/06/91
C
C UPDATE:  06/12/91 - PE BRIDEN: 'NSTORE' INCREASED FROM 10 TO 100
C
C UPDATE:  28/02/92 - PE BRIDEN: 'NSTORE' INCREASED FROM 100 TO 160
C
C UPDATE:  10/03/93 - PE BRIDEN: INTRODUCED CALL TO XXUID TO ESTABLISH
C                                IF USERID OF INPUT DATASET CHANGES
C                                BETWEEN CALLS.
C                                SAVE NAME OF LAST READ DATASET.
C                                (ADDED VARIABLES UIDIN,UIDLST,DSNREQ)
C
C UPDATE:   2/09/93 - HPS      : INTRODUCED CALL TO XXSSZD TO ESTABLISH
C                                IF USRGRP, USRTYP AND USREXT OF INPUT
C                                DATASET CHANGES BETWEEN CALLS.
C                                SAVE NAME OF LAST READ DATASET.
C                                (ADDED VARIABLES GRPIN,GRPLST,TYPIN,
C                                 TYPLST, EXTIN, EXTLST)
C
C UPDATE:   10/11/94 - L. JALOTA: MODIFIED TO RUN UNDER UNIX, SIZE OF
C   DSNAME AND DSNREQ INCREASED TO 80
```

```
C   CHARACTERS
C
C UPDATE:   21/11/94 - L/ JALOTA: TIDIED UP CHARACTER LENGTHS.
C
C UNIX-IDL PORT:
C
C DATE: UNKNOWN
C AUTHOR: UNKNOWN
C
C VERSION: 1.2 DATE: 08-07-96
C MODIFIED: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C     INCREASED NTDIM TO 35 IN LINE WITH THE REST OF 502
C
C VERSION: 1.3 DATE: 20-09-99
C MODIFIED: RICHARD MARTIN
C INCREASED TITLX TO CHAR*120
C
C VERSION: 1.4 DATE: 20-07-07
C MODIFIED: Allan Whiteford
C Small modification to comments to allow for automatic
C documentation preparation.
C
C VERSION: 1.5                          DATE: 26-03-08
C MODIFIED: Allan Whiteford
C              - Changed call from E2DATA to XXDATA_07
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      CHARACTER*120       TITLX
      INTEGER             IBSEL,       IRCODE,      ITVAL,       IZ
      INTEGER             IZ0IN,       IZ1,         METF,        METI
      LOGICAL             LTRNG(ITVAL)
      REAL*8              BWNO,        SZDA(ITVAL), TVAL(ITVAL)
```

7 Subroutine library adas7xx

7.1 g3astj: Subroutine g3astj from library adas7xx

```
C
       SUBROUTINE G3ASTJ(IWR04  , IWRP  , LPAPER , IOPT   , DATE ,
     &                   IZ0    , IZ    , IZ1    , NELEC  ,
     &                   BWNO   , MAXT  , TEA    ,
     &                   NAUTO  , ITIA  , ITCA   , AUGA   ,
     &                   NCONFG , ICF   , NLSETA , EPSIL  , EPSILMAX ,
     &                   NLEVL  , NFREE , IBREFA ,
     &                   IA     , ISA   , ILA    , ICFA   , EFREE    ,
     &                   ENERA  , WTA   , IXREFA , IFREFA ,
     &                   LPA    , ISPA  , XJP    ,
     &                   NTRAN  , IIPA  , IIA    , AA     , FA)

C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: G3ASTJ *********************
C
C  PURPOSE: CALCULATES COLLISION STRENGTHS IN IMPACT PARAMETER
C           APPROXIMATION AND WRITES THE ADF04 AND PAPER.TXT FILES.
C
C  CALLING PROGRAM: ADAS703
C
C  SUBROUTINE:
C
C
C
C  INPUT : (I*4)   NDTIN    = MAX. NUMBER OF INPUT TEMPERATURE/DENSITY
C          (I*4)   BWNO(I)  =  IONISATION POTENTIALS  (CM-1)
C                  TRA(IT)  =  REDUCED TEMPERATURE SET (K)
C                  MAXT     =  NUMBER OF TEMPERATURES  (LE.NDTIN)
C                  IOPT     =  OUTPUT CONFIGURATION FORM
C                              (1=EISSNER,0=STANDARD)
C
C                  IA()     =  LEVL INDEX
C                  ILA()    =  TOTAL ORBITAL QUANTUM NUMBER
C                  ISA()    =  TOTAL SPIN    QUANTUM NUMBER (2*S+1)
C                  IJA()    =  TOTAL ANGUL.  QUANTUM NUMBER (2*J)
C                  ICFA()   =  CONFIGRATION INDEX OF LEVL
C                  ENERA()  =  LEVEL ENERGY (CM-1)
C                  WTA()    =  (LEVL STATISTICAL WEIGHT-1)/2
C                  NLEVL    =  NUMBER OF LEVLS
C                  IIA()    =  LOWER LEVL INDEX OF TRANSITION
C                  IIPA()   =  UPPER LEVL INDEX OF TRANSITION
C                  AA()     =  A-VALUE FOR TRANSITION
C                  GAMSSA(,)=  GAMMAS FOR TRANSITION
C                  NTRAN    =  NUMBER OF TRANSITIONS
C
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C    XXFLSH    IDL-ADAS CALLS FLUSH TO CLEAR PIPES.
C
C
C AUTHOR   : Martin O'Mullane
C
C DATE     : 29/04/98
C
```

```
C VERSION  : 1.1                          DATE: 29-04-98
C MODIFIED : Martin O'Mullane
C            FIRST VERSION
C
C-----------------------------------------------------------------------
      CHARACTER*8         DATE
      INTEGER             IA(NDLEVL),  IBREFA(NDLEVL)
      INTEGER             ICF(NDCFG),  ICFA(NDLEVL)
      INTEGER             IFREFA(NDLEVL),           IIA(NDTRAN)
      INTEGER             IIPA(NDTRAN),             ILA(NDLEVL), IOPT
      INTEGER             ISA(NDLEVL), ISPA(NDFREE)
      INTEGER             ITCA(NDTRAN),             ITIA(NDTRAN)
      INTEGER             IWR04,       IWRP,        IXREFA(NDLEVL)
      INTEGER             IZ,          IZ0,         IZ1
      INTEGER             LPA(NDFREE), MAXT,        NAUTO,       NCONFG
      INTEGER             NELEC,       NFREE,       NLEVL
      INTEGER             NLSETA(NDCFG,30,2),       NTRAN
      LOGICAL             LPAPER
      REAL*8              AA(NDTRAN),  AUGA(NDTRAN)
      REAL*8              BWNO(10),    EFREE(NDFREE)
      REAL*8              ENERA(NDLEVL),            EPSIL(15)
      REAL*8              EPSILMAX,    FA(NDTRAN),  TEA(NDTIN)
      REAL*8              WTA(NDLEVL), XJP(NDFREE)
```

7.2 g3chip: Subroutine g3chip from library adas7xx

```
C
      SUBROUTINE G3CHIP( NCONFG , NELEC , NLSETA , ICF ,
     &                   IPSTR  , NP    )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: G3CHIP *********************
C
C  PURPOSE: TO SETUP A STRING CONTAINING THE NL OF THE PARENT
C           CONFIGURATION FOR INPUT BY THE USER.
C
C  CALLING PROGRAM: ADAS703
C
C  SUBROUTINE:
C
C  INPUT:  (I*4)  NCONFG  = PARENT CONF. STRING
C  INPUT:  (I*4)  NELEC   = PARENT CONF. STRING
C  INPUT:  (I*4)  NLSETA()= PARENT CONF. STRING
C  INPUT:  (I*4)  ICF()   = CONFIGRATION INDEX OF LEVEL
C
C  OUTPUT: (C*8)  IPSTR() = PARENT CONF. STRING
C
C  OUTPUT: (I*4)  NP      = NO. PARENTS
C
C
C AUTHOR:  Martin O'Mullane
C
C DATE:    29/04/98
C
C VERSION : 1.1 DATE: 01-05-98
C MODIFIED: Martin O'Mullane
C     - FIRST VERSION.
C
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      CHARACTER*9         IPSTR(NDP)
      INTEGER             ICF(NDCFG),  NCONFG,      NELEC
      INTEGER             NLSETA(NDCFG,NDCFG,2),    NP
```

7.3 g4llev: Subroutine g4llev from library adas7xx

```
      SUBROUTINE G4LLEV( LVA , LVF , NLVB, IND, NLVF, INDF , IFLAG )

      IMPLICIT REAL*8(A-H,O-Z)
C-----------------------------------------------------------------------
C
C******************* FORTRAN77 ROUTINE: G4LLEV **********************
C
C  PURPOSE:  Determines whether a particular transition belongs
C            to the required set by checking the bound and free
C            states against the level lists.
C
C       INPUT:
C          LVA    -   TEST BOUND LEVEL INDEX
C          LVF    -   TEST FREE LEVEL INDEX
C          NLVB   -   NO. OF BOUND STATES
C          IND()  -   SET OF BOUND STATES
C          NLVF   -   NO. OF FREE STATES
C          INDF() -   SET OF FREE STATES
C
C       OUTPUT:
C          IFLAG  -   = 0, LVA AND LVF DO NOT BELONG TO IND, INDF
C                     = 1, LVA AND LVF DO     BELONG TO IND, INDF
C
C
C  VERSION  : 1.1
C  DATE     : 15/09/2000
C  MODIFIED : Martin O'Mullane
C             First version.
C
C  VERSION  : 1.2
C  DATE     : 16/05/2007
C  MODIFIED : Martin O'Mullane
C             - Updated comments as part of subroutine
C               documentation production.
C
C-----------------------------------------------------------------------

      INTEGER             IFLAG,       IND(NDLEV),  INDF(NDLEV), LVA
      INTEGER             LVF,         NLVB,        NLVF
```

7.4 g4ppnrb: Subroutine g4ppnrb from library adas7xx

```
C
      subroutine g4ppnrb(iunt09  , isreq  , wghtn  , emin , emax ,
     &                   iunt10  , iunt07 , ltext  , date , user )

C-----------------------------------------------------------------------
C
C  ******************* FORTRAN77 ROUTINE: G4PPNRB **********************
C
C  PURPOSE: Program to post-process AUTOSTRUCTURE to give the spin
C           breakdown Auger rates.
C
C
C-----------------------------------------------------------------------
C
C  Based upon PPAUTS2:
C
C  PROGRAM PPAUTS2
C  ---------------
C
C  PROGRAM TO READ OUTPUT FILE FROM AUTOSTRUCTURE AND STRIP OUT
C  AUTOIONISATION TRANSITION PROBABILITIES OF INTEREST
C
C  REVISED DATA READING ALGORITHM
C
C  THIS VERSION OF CODE INTERROGATES MULTIPLE N AND L OUTPUT AS
C  PREFERRED BY NRB.
C
C  THUS REQUIRES A PRIOR RUN OF AUTOSTRUCTURE WITH OUTPUT GENERALLY
C  WRITTEN TO A FILE SUCH AS
C              JETUID.AS91#C.AS#O1.PASS3
C
C  CHOICES ON THE SPIN (2S+1) OF THE BOUND STATE AND ON THE ENERGY OF
C  THE FREE STATE ENABLE RESOLUTION INTO SPIN SYSTEM AND PARENT
C
C  THE FINAL OUTPUT CONTAINS WEIGHTED SUMS OF AUGER TRANSITION
C  PROBABILITIES, WHICH ARE SUMMED OVER L SEPARATELY AND THEN OVER N.
C
C  EACH INDIVIDUAL AUGER TRANSITION PROBABILITY IS WEIGHTED BY (2J+1)
C
C  AN N-SHELL AVERAGE IS OBTAINED BY NORAMILISING BY THE WEIGHT OF THE
C  ENTIRE N-SHELL. THIS IS INPUT BY THE USER AS A MULTIPLIER ON N**2
C         I.E.
C           N-SHELL WEIGHT = WGHTN * N**2
C
C
C   AUTHOR:  WILLIAM J. DICKSON
C   DATE:  2ND DECEMBER 1992
C   ADDRESS:  K1/1/26, JET JOINT UNDERTAKING
C
C-----------------------------------------------------------------------
C  ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          G4LLVL     ADAS      CHECK WETHER A SPECIFIC TRANSITION
C                               BELONGS TO REQUIRED SET
C
C
C  AUTHOR : Martin O'Mullane,
C           K1/1/43,
```

```
C           JET
C
C
C  VERSION  : 1.1
C  DATE     : 15/09/2000
C  MODIFIED : Martin O'Mullane
C             First version.
C
C
C-----------------------------------------------------------------------
      CHARACTER*8         DATE
      CHARACTER*30        USER
      INTEGER             ISREQ,       IUNT07,      IUNT09,      IUNT10
      LOGICAL             LTEXT
      REAL*8              EMAX,        EMIN,        WGHTN
```

7.5 gximpr: Subroutine gximpr from library adas7xx

```
C
       SUBROUTINE GXIMPR(IZ,WI,EI,WJ,EJ,M,AJI,EPS,OMEG,N,T,RAT,QI,QJ,
     &GA)
       IMPLICIT REAL*8(A-H,O-Z)
C-----------------------------------------------------------------------
C  PURPOSE: CALCULATES ELECTRON COLLISIONAL EXCITATION AND DEEXCITATION
C           RATE COEFFICIENTS FOR DIPOLE TRANSITIONS IN THE IMPACT
C           PARAMETER APPROXIMATION
C
C   (BURGESS AND SUMMERS,1976,MON.NOT.R.AST.SOC.,174,345)
C
C   OPTIONALLY A SET OF INCIDENT ELECTRON ENERGIES AND COLLISION STRENGT
C   MAY BE PROVIDED, IN WHICH CASE THE IMPACT PARAMETER THEORY IS USED T
C   CALCULATE THE COLLISION STRENGTHS AT HIGH ENERGY WITH VALUES SCALED
C   THE HIGHEST ENERGY INPUT COLLISION STRENGTH.
C   EITHER THE ABSORPTION OSCILLATOR STRENGTH OR THE EINSTEIN COEFFICIEN
C   MUST BE PROVIDED, THE OTHER BEING COMPUTED.
C  ARGUMENTS
C   IZ,WI,EI,WJ,EJ,M,AJI,EPS(20),OMEG(20),N,T(40),RAT,QI(40),QJ(40),
C   GA(40)
C  SUBROUTINES
C   EIQIP,XIP,YIP,ZERO1
C  INPUT
C       IZ=ION CHARGE
C       TRANSITION NAME TAKES THE FORM
C          ELECTRON TRANS.(COLS 11-15), ANGULAR TRANS.(COLS 21-40)
C       WI=STATISTICAL WEIGHT OF STATE I
C       EI=BINDING ENERGY OF STATE I (RYDBERGS)
C       WJ=STATISTICAL WEIGHT OF STATE J
C       EJ=BINDING ENERGY OF STATE J (RYDBERGS)
C       M=NUMBER OF TABULAR VALUES OF COLLISION STRENGTH
C       FIJ=ABSORPTION OSCILLATOR STRENGTH FOR TRANSITION
C       AJI=EINSTEIN COEFFICIENT FOR TRANSITION
C       EPS(K)=INCIDENT ELECTRON ENERGIES (RYDBERGS)
C       OMEG(K)=COLLISION STRENGTHS
C       N=NUMBER OF ELECTRON TEMPERATURES
C       T(I)=ELECTRON TEMPERATURES (DEGS. K)
C  OUTPUT
C   RAT=RATIO OF OMEG(M) TO I.P. OMEGA.
C   QI(I)=COLLISIONAL EXCITATION RATE COEFFICIENTS
C   QJ(I)=COLLISIONAL DEEXCITATION RATE COEFFICIENTS.
C   GA(I)=GAMMA PARAMETER
C  AUTHOR
C   HUGH SUMMERS     1977/5/20
C  UPDATES
C       1983/9/1 1985/6/13  ***** LATTER IS IMPORTANT CORRECTION
C                                 IDENTIFIED IN CODE
C  COMMENTS
C   I IS THE LOWER LEVEL OF THE TRANSITION.
C   M MAY BE ZERO, IN WHICH CASE NO EPS AND OMEG VALUES ARE REQUIRED.
C   UNDERFLOW IS NOT TRAPPED. THIS MAY BE ACHEIVED IN IBM FORTRAN WITH T
C
C  VERSION: 1.1 DATE: 18-06-98
C  MODIFIED: MARTIN OMULLANE
C  - INCLUDED IN ADAS.
C
C  VERSION: 1.2 DATE: 18-06-98
C  MODIFIED: ALLAN WHITEFORD
C  - UPDATED COMMENTS AS PART OF SUBROUTINE DOCUMENTATION
```

```
C                  PROCEDURE
C
C-----------------------------------------------------------------------
      INTEGER             IZ,           M,            N
      REAL*8              AJI,          EI,           EJ,           EPS(20)
      REAL*8              GA(40),       OMEG(20),     QI(40),       QJ(40)
      REAL*8              RAT,          T(40),        WI,           WJ
      REAL*8              EI,           EIJ,          EIQ,          EM
      REAL*8              FLAG,         PHI,          R,            SC
      REAL*8              WI,           WJ,           Z
      REAL*8              DELTA,        XI
      REAL*8              DELTA,        XI
      REAL*8              A,            B,            D1,           E
      REAL*8              T2,           TIF,          VA,           VB
      REAL*8              X,            XI,           Z
```

8 Subroutine library adas8xx

8.1 diag: Subroutine diag from library adas8xx

```
C
      SUBROUTINE DIAG(N,Z,D)
      IMPLICIT REAL*8(A-H,O-Z)
C
C PURPOSE: DIAGONALIZATION OF REAL SYMMETRIC N-BY-N MATRIX Z.
C
C INPUT REQUIRED. N AND Z. ONLY LOWER TRIANGLE OF Z NEED BE SUPPLIED.
C                          MATRIX Z OVERWRITTEN BY EIGENVECTORS OF Z.
C OUTPUT.         Z AND D, WHERE Z CONSISTS OF COLUMN EIGENVECTORS
C                          AND D CONSISTS OF CORRESPONDING EIGENVALUES.
C N.B. THE VALUE OF N MUST NOT EXCEED THE DIMENSION OF THE VARIABLE E,
C SPECIFIED IN THE FOLLOWING LINE.
C
C-----------------------------------------------------------------------
C
C  VERSION  : 1.1
C  DATE     : ?
C  MODIFIED : H P Summers
C              - Initial version.
C
C  VERSION  : 1.2
C  DATE     : 16-05-2007
C  MODIFIED : Allan Whiteford
C              - Remove listing information from colums 72+.
C              - Updated comments as part of subroutine documentation
C          procedure.
C-----------------------------------------------------------------------
      INTEGER             N
      REAL*8              D(10),       Z(10,10)
```

8.2 dipsum: Subroutine dipsum from library adas8xx

```
C
      REAL FUNCTION DIPSUM*8(JZ,L,E1,E2)
      IMPLICIT REAL*8(A-H,O-Z)
C
C PURPOSE: Calculates a Burgess Dipole Sum.
C
C CALCULATES THE SUM GIVEN IN EQUATIONS (10) AND (11) OF A. BURGESS,
C J. PHYS. B7,?,1974. SET JZ TO ZERO FOR THE ZERO CHARGE (NEUTRAL
C ATOM) CASE. SET L TO THE LOWER LIMIT OF SUMMATION (MUST BE GREATER
C THAN ZERO). SET E1=(KAPPA1)**2 FOR NON ZERO CHARGE, =(K1)**2
C FOR ZERO CHARGE. SET E2=(KAPPA2)**2 FOR NON ZERO CHARGE,
C =(K2)**2 FOR ZERO CHARGE.
C-----------------------------------------------------------------------
C
C  VERSION  : 1.1
C  DATE     : ?
C  MODIFIED : H P Summers
C              - Initial version.
C
C  VERSION  : 1.2
C  DATE     : 16-05-2007
C  MODIFIED : Allan Whiteford
C              - Remove listing information from colums 72+.
C              - Updated comments as part of subroutine documentation
C          procedure.
C-----------------------------------------------------------------------
      L1=L
      L2=L-1
      IF(JZ)1,2,1
1     F1=R8FDIP(E1,L1,E2,L2)
      F2=R8FDIP(E1,L2,E2,L1)
      EL=L
      DIPSUM=(F1-F2)*(F1+F2)*(1.0+EL*EL*E1)/(EL*(E1-E2))
      RETURN
2     F1=R8FDIP0(E1,L1,E2,L2,1.0D-12)
      F2=R8FDIP0(E1,L2,E2,L1,1.0D-12)
      EL=L
      DIPSUM=(F1-F2)*(F1+F2)*EL*E1/(E1-E2)
      RETURN
      END
```

8.3 fcf4: Subroutine fcf4 from library adas8xx

```
C
       subroutine fcf4(f,c,x0,e,z,el,x1,h)
C-----------------------------------------------------------------------
C
C  ****************** fortran77 program: fcf4.for **********************
C
C  Purpose:  Evaluates free regular Coulomb real function
C
C            Puts result in
C            f(j),j=1,2,...,x1/h.
C            f satisfies ((d/dx)(d/dx)-el(el+1)-2z/x+e)f=0
C            f=c*x**(el+1.0)*(1.0+...)   for small x
C            f=k**(-0.5)*dsin(kx-0.5*el*pi-(z/k)log(2kx)+
C            arggamma(el+1+i*z/k))  for large x
C            where k=dsqrt(e)
C            n.b. z is positive for repulsive field
C
C  Subroutine:
C
C  input : (r*8)  e        = energy (Ryd)
C  input : (r*8)  z        = effective charge seen by electron
C  input : (r*8)  el       = orbital angular momentum
C  input : (r*8)  x1       = outer limit for tabulation
C  input : (r*8)  h        = tabulation step length
C
C  output: (r*8)  f()      = resulting Coulomb function
C  output: (r*8)  c        = normalisation constant
C  output: (r*8)  x0       = the (approx) first point of inflexion in f
C
C           (r*8) wilf      = fortran function
C
C
C  Routines:
C           none
C
C  Author:  H. P. Summers, University of Strathclyde
C            ja7.08
C            tel. 0141-548-4196
C
C  Date:   24/02/03
C
C  Update: HP Summers   24/05/04 Restructure and addded standard warning
C
C  Update: AD Whiteford 20/07/07 Modified comments slightly to allow
C                                for automatic generation of
C                                documentation.
C
C-----------------------------------------------------------------------
      REAL*8              C,           E,           EL,          F(1000)
      REAL*8              H,           X0,          X1,          Z
```

8.4 h4angf: Subroutine h4angf from library adas8xx

```
C
       subroutine h4angf( xsp   , xlp   , xl0   , xst0  , xlt0  ,
     &                    xl1   , xst1  , xlt1  , xnq   , fpc   ,
     &                    xjp   , xj0   , xjt0  , xj1   , xjt1  ,
     &                    anga  , lama  , nlam
     &                )
c-----------------------------------------------------------------------
c
c  ****************** fortran77 subroutine: h4angf *********************
c
c  purpose:  calculates angular factors for born approximation
c
c
c  subroutine:
c
c  input : (r*8)  xsp       = 2*Sp+1 parent spin angular momentum
c  input : (r*8)  xlp       = Lp parent orbital angular momentum
c  input : (r*8)  xl0       = l initial valence electron orbital
c  input : (r*8)  xst0      = 2*S+1  total initial spin anular momentum
c  input : (r*8)  xlt0      = L  total initial orbital angular momentum
c  input : (r*8)  xl1       = l' final valence electron orbital
c  input : (r*8)  xst1      = 2*S'+1  total initial spin anular momentum
c  input : (r*8)  xlt1      = L'  total initial orbital angular momentum
c  input : (r*8)  xnq       = number of equivalent electrons in initial
c                             state
c  input : (r*8)  fpc       = fractional parentage coefficient
c  input : (r*8)  xjp       = Jp  parent total angular momentum
c  input : (r*8)  xj0       = j  initial valence electron angular mom.
c  input : (r*8)  xjt0      = J  total initial angular momentum
c  input : (r*8)  xj1       = j'  final valence electron angular mom.
c  input : (r*8)  xjt1      = J'  total initial angular momentum
c
c  output: (r*8)  anga()   = angular factor for each multipole
c  output: (i*4)  lama()   = multipole
c  output: (i*4)  nlam     = number of multipoles in anga vector
c
c
c  output: (c*80) cstrg     = string marking end of partition block
c
c  routines:
c          routine    source    brief description
c          ------------------------------------------------------------
c          wig3j      adas      evaluates the wigner 3-j symbol
c          wig6j      adas      evaluates the wigner 6-j symbol
c          i4unit     adas      fetch unit number for output of messages
c
c  author:  h. p. summers, university of strathclyde
c           ja7.08
c           tel. 0141-548-4196
c
c  date:    24/02/03
c
c  update:
c
c-----------------------------------------------------------------------
c-----------------------------------------------------------------------
      INTEGER             LAMA(20),    NLAM
      REAL*8              ANGA(20),    FPC,         XJ0,         XJ1
      REAL*8              XJP,         XJT0,        XJT1,        XL0
```

```
      REAL*8           XL1,         XLP,         XLT0,        XLT1
      REAL*8           XNQ,         XSP,         XST0,        XST1
```

8.5 h4born: Subroutine h4born from library adas8xx

```
C
       subroutine h4born( ixdim  , itdim   , isdim  , ndinfo ,
     &                    cameth ,
     &                    z0     , z       , zeff   ,
     &                    n1     , l1      , eb1    ,
     &                    n2     , l2      , eb2    ,
     &                    isp    , lp      , xjp    ,
     &                    ist1   , lt1     , xj1    , xjt1   ,
     &                    ist2   , lt2     , xj2    , xjt2   ,
     &                    neqv1  , fpc1    , neqv2  , fpc2   ,
     &                    aval   ,
     &                    xmax   , iext    , ijucys , jealfa ,
     &                    nshell , nc      , lc     , numel  , alfaa  ,
     &                    nx     , xa      , omga   ,
     &                    nt     , tea     , upsa   ,
     &                    cresol , ixtyp   , s12    , e12    ,
     &                    ninfo  , cinfoa
     &                  )
C-----------------------------------------------------------------------
C
C  ***************** fortran77 program: h4born.for *********************
C
C  Purpose:  Calculation of Born cross-sections using numerical wave
C            functions.
C
C
C  Subroutine:
C
C  input : (i*4)  ixdim    = maximum dimension for X  array
C  input : (i*4)  itdim    = maximum dimension for Te array
C  input : (i*4)  isdim    = maximum dimension for shell vectors
C  input : (i*4)  ndinfo   = maximum number of information strings
C
C  input : (c*1)  cameth   = the tag distinguishing the type of
C                            analysis: a - Born, b- IP
C  input : (r*8)  z0       = nuclear charge
C  input : (r*8)  z        = ion charge
C  input : (r*8)  zeff     = ion charge +1
C  input : (i*4)  n1       = lower n-shell of transition
C  input : (i*4)  l1       = lower l-shell of transition
C  input : (r*8)  eb1      = binding energy (Ryd) in lower level
C  input : (i*4)  n2       = upper n-shell of transition
C  input : (i*4)  l2       = upper l-shell of transition
C  input : (r*8)  eb2      = binding energy (Ryd) in upper level
C  input : (i*4)  isp      = 2*Sp+1 for parent
C  input : (i*4)  lp       = Lp for parent
C  input : (r*8)  xjp      = Jp for parent (if 'ic' coupling)
C  input : (i*4)  ist1     = 2*S+1 for lower state
C  input : (i*4)  lt1      = L for lower state
C  input : (r*8)  xj1      = j for lower state
C  input : (r*8)  xjt1     = J for lower state
C  input : (i*4)  ist2     = 2*S'+1 for upper state
C  input : (i*4)  lt2      = L' for upper state
C  input : (r*8)  xj2      = j' for lower state
C  input : (r*8)  xjt2     = J' for upper state
C  input : (i*4)  neqv1    = no. of equiv. electrons for lower shell.
C  input : (r*8)  fpc1     = fract. parentage for lower state
C  input : (i*4)  neqv2    = no. of equiv. electrons for upper shell.
C  input : (r*8)  fpc2     = fract. parentage for upper state
```

```
C  input : (i*4)  aval     = A-value (sec-1) if dipole; else -ve
C  input : (i*4)  xmax     = range of numerical wave functions
C  input : (i*4)  iext     = 0 => calculate radial wave functions
C                           = 1 => read in radial wave functions
C  input : (i*4)  ijucys   = 0  => Slater potential form adopted
C                           = 1 => Jucys potential form adopted
C  input : (i*4)  jealfa   = 0 => fcf6 search for energy eigenvalue
C                           = 1 => fcf6 search for scaling parameters
C  input : (i*4)  nshell   = number of screening shells
C  input : (i*4)  nc()     =  n for each screening shell
C                              1st dim: screening shell index
C  input : (i*4)  lc()     = l for each screening shell
C                              1st dim: screening shell index
C  input : (i*4)  numel()  = iq for each screening shell
C                              1st dim: screening shell index
C  input : (r*8)  alfaa    = scaling factor for each screening shell
C                              1st dim: index for lower & upper state
C                              2nd dim: index over screening shells
C  input : (i*4)  nx       = number of incident electron energies
C  input : (r*8)  xa()     = threshold parameter values
C  input : (i*4)  nt       = number of electron temperatures
C  input : (r*8)  tea()    = electron temperatures (K)
C  input : (c*2)  cresol   = resolution level 'ls' or 'ic'

C  output: (r*8)  omga()   = omegas at input values of x-parameter
C  output: (r*8)  upsa()   = upsilon at onput values of Te
C  output: (i*4)  ixtyp    = transition type 1=dipole;2=non-dipole;
C                                           3=spin-change
C  output: (r*8)  s12      = line strength (dipole) otherwise zero
C  output: (r*8)  e12      = level energy difference (Ryd)
C  output: (i*4)  ninfo    = number of information strings
C  output  (c*90) cinfoa() = information strings
C                            1st dim: index number of strings
C
C        : (i*4)  icount    = general counter index
C        : (i*4)  iq        = general num. of equiv. electrons
C        : (i*4)  irept     = 0 => full wave function determination
C                           = 1 => use wave functions from previous case
C        : (i*4)  iswit     = 1 => no spin change
C                           = 2 => spin change
C        : (i*4)  j         = general integer variable
C        : (i*4)  j1        = general integer variable
C        : (i*4)  jalf1     = lower range of scaling parameters
C        : (i*4)  jalf2     = upper range of scaling parameters
C        : (i*4)  jh        = radial wave function tabulation size
C        : (i*4)  jsn       = -1 => Jucys potential form adopted
C                           = 0  => Slater potential form adopted
C        : (i*4)  lam       = multipole number
C        : (i*4)  n         = general principal quanum number
C        : (i*4)  nlam      = number of multipoles in anga vector
C        : (i*4)  nq        = number of equivalent electrons in initial
C                             state
C        : (r*8)  acc       = precision for eigenvalue search
C        : (r*8)  de        = transition energy (ryd)
C        : (r*8)  fpc       = fractional parentage coefficient
C        : (r*8)  ei        = incid. electron energy (ryd)
C        : (r*8)  eiu       = adj. incid. engy. for Cowan threshold (ryd)
C        : (r*8)  h         = step length for radial wave fn. tabulation
C        : (r*8)  omegb     = Born cross-section
C        : (r*8)  q         = number of equivalent electrons in initial
C                             state
```

```
C        : (r*8)  res       = general result variable
C        : (r*8)  t         = general real variable
C        : (r*8)  xlp       = Lp parent orbital angular momentum
C        : (r*8)  xl1       = l initial valence electron orbital
C        : (r*8)  xlt1      = L  total initial orbital angular momentum
C        : (r*8)  xl2       = l' final valence electron orbital
C        : (r*8)  xlt2      = L'  total initial final angular momentum
C        : (r*8)  xn1       = principal quantum number of initial shell
C        : (r*8)  xn2       = principal quantum number of final shell
C        : (r*8)  xnq       = number of equivalent electrons in initial
C        : (r*8)  xsp       = 2*Sp+1 parent spin angular momentum
C        : (r*8)  xst1      = 2*S+1  total initial spin anular momentum
C        : (r*8)  xst2      = 2*S'+1  total final spin anular momentum
C                             state
C        : (r*8)  z0        = nuclear charge
C        : (r*8)  z1        = ion charge + 1
C        : (r*8)  z         = ion charge
C        : (r*8)  zeff      = effective ion charge
C        : (r*8)  zz0       = -z0
C
C        : (i*4)  nlqs()    = 1000*n+100*l+iq for each screening shell
C                              1st dim: screening shell index
C        : (i*4)  na()      = princ. qu. no. for val. elec.(initial & final)
C        : (i*4)  la()      = l qu. no. for val. elec.(initial & final)
C        : (i*4)  lama()    = multipole value
C
C        : (r*8)  alfa()    = scaling factor for each screening shell
C                              1st dim: index over screening shells
C        : (r*8)  anga()    = angular factor for each multipole
C                              1st dim: index over included multipoles
C        : (r*8)  ea()      = energy (ryd) for active electron.  NB -ve for
C                             a bound state
C                              1st dim: index for initial & final state
C        : (r*8)  omega()   = collision strength for each multipole
C                              1st dim: index over included multipoles
C        : (r*8)  qda()     = quantum defect for valence electron.
C                              1st dim: index for initial & final state
C        : (r*8)  x0a()     = inner turning pt. for val. elec. wave fn.
C                              1st dim: index for initial & final state
C        : (r*8)  x1a()     = outer turning pt. for val. elec. wave fn.
C                              1st dim: index for initial & final state
C        : (r*8)  x2a()     = range for active elec. wave fn.
C                              1st dim: index for initial & final state
C
C
C  Routines:
C          routine    source    brief description
C          ------------------------------------------------------------
C          gamaf      adas      tabulates factorials
C          h4angf     adas      evaluates the Born angular parts
C          rdwbes     adas      evaluates the Born radial parts
C          i4unit     adas      fetch unit number for output of messages
C
C  Author:  H. P. Summers, University of Strathclyde
C           ja7.08
C           tel. 0141-548-4196
C
C  Date:   25/02/03
C
C  Update: HP Summers  21/05/04  Restructure and add calculation
C                                information strings to parameter output.
```

```
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      CHARACTER           CAMETH
      CHARACTER*90        CINFOA(NDINFO)
      CHARACTER*2         CRESOL
      INTEGER             IEXT,         IJUCYS,       ISDIM,        ISP
      INTEGER             IST1,         IST2,         ITDIM,        IXDIM
      INTEGER             IXTYP,        JEALFA,       L1,           L2
      INTEGER             LC(ISDIM),    LP,           LT1,          LT2
      INTEGER             N1,           N2,           NC(ISDIM),    NDINFO
      INTEGER             NEQV1,        NEQV2,        NINFO,        NSHELL
      INTEGER             NT,           NUMEL(ISDIM),               NX
      REAL*8              ALFAA(2,ISDIM),             AVAL,         E12
      REAL*8              EB1,          EB2,          FPC1,         FPC2
      REAL*8              OMGA(IXDIM), S12,           TEA(ITDIM)
      REAL*8              UPSA(ITDIM), XA(IXDIM),     XJ1,          XJ2
      REAL*8              XJP,          XJT1,         XJT2,         XMAX
      REAL*8              Z,            Z0,           ZEFF
```

8.6 h4data: Subroutine h4data from library adas8xx

```
C
       subroutine h4data( iunit  , ixdim  , itdim   , isdim  ,
     &                    dsfull , indxref , title  , cameth ,
     &                    z0     , z       , zeff   ,
     &                    n1     , l1      , eb1    ,
     &                    n2     , l2      , eb2    ,
     &                    isp    , lp      , xjp    ,
     &                    ist1   , lt1     , xj1    , xjt1   ,
     &                    ist2   , lt2     , xj2    , xjt2   ,
     &                    neqv1  , fpc1    , neqv2  , fpc2   ,
     &                    aval   ,
     &                    xmax   , iextwf  , ijucys , isrch  ,
     &                    nshell , nlqs    , alfaa  ,
     &                    nx     , xa      ,
     &                    nt     , tea     ,
     &                    cresol
     &                  )
c-----------------------------------------------------------------------
c
c  ******************** fortran77 subroutine h4data ******************
c
c  purpose: to refresh a data index from an adas804 archive. reads
c           in the index code a-effective potential Born, b-impact
c           parameter and the the rest of the data as appropriate.
c
c  calling program: adas804
c
c  subroutine:
c
c  input : (i*4)  iunit    = unit to be used for reading file
c  input : (i*4)  ixdim    = maximum dimension for X  array
c  input : (i*4)  itdim    = maximum dimension for Te array
c  input : (i*4)  isdim    = maximum dimension for shell vectors
c
c  input : (c*80) dsfull   = the users' chosen archive file name.
c  input : (i*4)  indxref  = the index number to refresh from.
c
c  output: (c*40) title    = transition title in the archive file.
c  output: (c*1)  cameth   = the tag distinguishing the type of
c                            analysis: a - Born, b- IP
c  output: (r*8)  z0       = nuclear charge
c  output: (r*8)  z        = ion charge
c  output: (r*8)  zeff     = ion charge +1
c  output: (i*4)  n1       = lower n-shell of transition
c  output: (i*4)  l1       = lower l-shell of transition
c  output: (r*8)  eb1      = binding energy (Ryd) in lower level
c  output: (i*4)  n2       = upper n-shell of transition
c  output: (i*4)  l2       = upper l-shell of transition
c  output: (r*8)  eb2      = binding energy (Ryd) in upper level
c  output: (i*4)  isp      = 2*Sp+1 for parent
c  output: (i*4)  lp       = Lp for parent
c  output: (r*8)  xjp      = Jp for parent (if 'ic' coupling)
c  output: (i*4)  ist1     = 2*S+1 for lower state
c  output: (i*4)  lt1      = L for lower state
c  output: (r*8)  xj1      = j for lower state
c  output: (r*8)  xjt1     = J for lower state
c  output: (i*4)  ist2     = 2*S'+1 for upper state
c  output: (i*4)  lt2      = L' for upper state
c  output: (r*8)  xj2      = j' for upper state
```

```
c  output: (r*8)  xjt2     = J' for upper state
c  output: (i*4)  neqv1    = no. of equiv. electrons for lower shell.
c  output: (r*8)  fpc1     = fract. parentage for lower state
c  output: (i*4)  neqv2    = no. of equiv. electrons for upper shell.
c  output: (r*8)  fpc2     = fract. parentage for upper state
c  output: (i*4)  aval     = A-value (sec-1) if dipole; else -ve
c  output: (i*4)  xmax     = range of numerical wave functions
c  output: (i*4)  iextwf   = 0 => calculate radial wave functions
c                          = 1 => read in radial wave functions
c  output: (i*4)  ijucys   = -1 => Jucys potential form adopted
c                          = 0  => Slater potential form adopted
c  output: (i*4)  isrch    = 0 => fcf6 search for energy eigenvalue
c                          = 1 => fcf6 search for scaling parameters
c  output: (i*4)  nshell   = number of screening shells
c  output: (i*4)  nlqs     = 1000*n+100*l+iq for each screening shell
c                              1st dim: screening shell index
c  output: (i*4)  alfaa    = scaling factor for each screening shell
c                              1st dim: index for lower & upper state
c                              2nd dim: index over screening shells
c  output: (i*4)  nx       = number of incident electron energies
c  output: (i*4)  xa()     = threshold parameter values
c  output: (i*4)  nt       = number of electron temperatures
c  output: (i*4)  tea()    = electron temperatures (K)
c  output: (i*4)  cresol   = 'ic' => transition between J-levels
c                          = 'ls' => transition between terms
c
c
c  routines:
c          routine    source    brief description
c          ------------------------------------------------------------
c          i4unit     adas      fetch unit number for output of messages
c          xxword     adas      extract position of number in buffer
c          xxcase     adas      change string to all upper or lower case
c
c  author:  Hugh P. Summers, University of Strathclyde
c           JA7.08
c           Tel.: +44 (0)141-548-4196
c
c  date:    24 February 2003
c
c
c  version: 1.1   Hugh Summers  24/02/03
c  modified:      first release
c
c-----------------------------------------------------------------------
      CHARACTER           CAMETH
      CHARACTER*2         CRESOL
      CHARACTER*80        DSFULL
      CHARACTER*40        TITLE
      INTEGER             IEXTWF,      IJUCYS,      INDXREF,     ISDIM
      INTEGER             ISP,         ISRCH,       IST1,        IST2
      INTEGER             ITDIM,       IUNIT,       IXDIM,       L1
      INTEGER             L2,          LP,          LT1,         LT2
      INTEGER             N1,          N2,          NEQV1,       NEQV2
      INTEGER             NLQS(ISDIM), NSHELL,      NT,          NX
      REAL*8              ALFAA(2,ISDIM),           AVAL,        EB1
      REAL*8              EB2,         FPC1,        FPC2
      REAL*8              TEA(ITDIM),  XA(IXDIM),   XJ1,         XJ2
      REAL*8              XJP,         XJT1,        XJT2,        XMAX
      REAL*8              Z,           Z0,          ZEFF
```

8.7 h4fasy: Subroutine h4fasy from library adas8xx

```
C
      subroutine h4fasy( istdim ,
     &                   x      , xa    , n    , ya    , y     , dy    ,
     &                   c1     , c2    , c3   , c4    ,
     &                   form   , iforms
     &                 )
c-----------------------------------------------------------------------
c
c  *************** fortran77 subroutine: h4spl3.for ******************
c
c  purpose: provide a spline interpolate making use of specified
c           asymptotic behaviour
c
c  calling program: various
c
c  notes: (1) uses labelled common  /espl3/
c
c  input : (i*4)  istdim  = dimensionality for splining arrays
c
c  input : (r*8)  x       = required x-value
c  input : (i*4)  x(i)    = knots
c  input : (r*8)  n       = number of knots
c
c  input : (r*8)  c1(i,j) = 1st spline coefficient precursor
c  input : (r*8)  c2(i,j) = 2nd spline coefficient precursor
c  input : (r*8)  c3(i,j) = 3rd spline coefficient precursor
c  input : (r*8)  c4(i,j) = 4th spline coefficient precursor
c  input : (r*8)  form    = external function specifying asymptotic forms
c  input : (i*4)  iforms  = index of required form
c
c  output: (r*8)  y=returned y-value
c  output: (r*8)  dy=returned derivative
c
c
c  routines:
c           routine    source    brief description
c           ------------------------------------------------------------
c           h4fspl     adas
c           i4unit     adas      fetch unit number for output of messages
c
c  author:  Hugh P. Summers, University of Strathclyde
c           JA7.08
c           Tel.: +44 (0)141-548-4196
c
c  date:    24 July 2002
c
c
c  version: 1.1   Hugh Summers  24/07/02
c  modified:      first release
c
c-----------------------------------------------------------------------
      INTEGER             IFORMS,      ISTDIM,      N
      REAL*8              C1(ISTDIM,ISTDIM-1),      C2(ISTDIM,ISTDIM-1)
      REAL*8              C3(ISTDIM,ISTDIM-1),      C4(ISTDIM,ISTDIM-1)
      REAL*8              DY,          X,           XA(ISTDIM),  Y
      REAL*8              YA(ISTDIM)
```

8.8 h4form: Subroutine h4form from library adas8xx

```
C
       function h4form(i,x)
c-----------------------------------------------------------------------
c
c  *************** fortran77 function: h4form.for ******************
c
c  purpose: specifies an independent variable transforma for splining
c
c  calling program: various
c
c  input : (i*4)  i       = selection parameter for transform
c  input : (r*8)  x       = independent variable
c
c  output: (r*8)  h4form  = transformed independent variable

c
c  routines:
c
c  author:  Hugh P. Summers, University of Strathclyde
c           JA7.08
c           Tel.: +44 (0)141-548-4196
c
c  date:    24 July 2002
c
c
c  version: 1.1   Hugh Summers  24/07/02
c  modified:      first release
c
c-----------------------------------------------------------------------
      INTEGER             I
      REAL*8              X
```

8.9 h4ftsp: Subroutine h4ftsp from library adas8xx

```
C
       subroutine h4ftsp( istdim  ,
     &                    x       , xa   , n    , yaa  , y    , dy    ,
     &                    i0      , c1   , c2   , c3   , c4   , isw
     &                  )
c-----------------------------------------------------------------------
c
c  ***************** fortran77 subroutine: h4ftsp.for *********************
c
c  purpose: obtain the value from a spline interpolation
c
c  calling program: various
c
c
c  input : (i*4)  istdim  = dimensionality for splining arrays
c
c  input : (r*8)  x       = required x-value
c  input : (r*8)  xa(i)   = x-values
c  input : (i*4)  n       = number of values
c  input : (r*8)  yaa(i)  = y-values (possibly stored as multiple sets)
c  input : (i*4)  i0      = starting index(-1) in yaa array of required input set
c  input : (r*8)  c1(i,j) = 1st spline coefficient precursor
c  input : (r*8)  c2(i,j) = 2nd spline coefficient precursor
c  input : (r*8)  c3(i,j) = 3rd spline coefficient precursor
c  input : (r*8)  c4(i,j) = 4th spline coefficient precursor
c  input : (i*4)  isw     = .le.0  ordinary     spline interpolation
c                         = .gt.0  logarithmic  spline interpolation
c
c  output: (r*8)  y       = returned y-value
c  output: (r*8)  dy      = returned derivative
c
c  routines:
c          routine    source    brief description
c          ------------------------------------------------------------
c          i4unit     adas      fetch unit number for output of messages
c
c  author:  Hugh P. Summers, University of Strathclyde
c           JA7.08
c           Tel.: +44 (0)141-548-4196
c
c  date:    24 July 2002
c
c
c  version: 1.1   Hugh Summers  24/07/02
c  modified:      first release
c
c-----------------------------------------------------------------------
      INTEGER             I0,          ISTDIM,      ISW,         N
      REAL*8              C1(ISTDIM,ISTDIM-1),      C2(ISTDIM,ISTDIM-1)
      REAL*8              C3(ISTDIM,ISTDIM-1),      C4(ISTDIM,ISTDIM-1)
      REAL*8              DY,          X,           XA(ISTDIM),  Y
      REAL*8              YAA(*)
```

8.10 h4gasy: Subroutine h4gasy from library adas8xx

```
C
       subroutine h4gasy( istdim  ,
     &                    x       , dx     ,
     &                    form    , iforms , iends
     &                   )
c-----------------------------------------------------------------------
c
c  ***************** fortran77 subroutine: h4gasy.for ******************
c
c  purpose: initialises common arrays required for splining with
c           smooth fitting to an asymptotic form
c
c  calling program: various
c
c  notes: (1) uses labelled common  /espl3/
c         (2) common /espl3/ is set by this routine
c         (3) conditions at 1st & nth nodes controlled by iend1 & iendn
c           iend = 1 : specified d log(y) ie. dy/y at node stored in
c                      appropriate vector
c                = 2 : zero curvature
c                = 3 : constant curvature
c                = 4 : matched to specified functional form in terms of
c                      two parameters a and b such that
c                        funct = p(1)*a+q(1)*b
c                        1st deriv. = p(2)*a+q(2)*b
c                        2nd deriv. = p(3)*a+q(3)*b
c                        where a1,b1,p1,q1 are used for 1st node and
c                        an,bn,pn,qn for nth node
c
c         (4) if iends=1,matching is at first knot(given by x)
c                     =2,matching is at  last knot(given by x)
c             asymptotic forms are given in external function form(i,x)
c             where i=4*iforms-5+2*iends points to 1st part of asymp. form
c                   =4*iforms-4+2*iends points to 2nd part of asymp. form
c             thus a series of asymptotic forms may be present in form
c
c
c  input : (i*4)  istdim  = dimensionality for splining arrays
c
c  input : (i*4)  n       = number of knots
c      common /espl3/ provides input in vector iend which specifies
c                     choice of end condition at first iend(1) or last
c                     iend(2) knot of spline
c  input : (r*8)  x       = x-value of end point
c  input : (r*8)  dx      = displacement from x-value for
c                           derivative evaluation
c  input : (r*8)  form    = external function specifying asymptotic forms
c  input : (i*4)  iforms  = selected form
c  input : (i*4)  iends   = 1,matching is at first knot(given by x)
c                         = 2,matching is at  last knot(given by x)
c
c
c  routines:
c           routine    source    brief description
c           ------------------------------------------------------------
c           i4unit     adas      fetch unit number for output of messages
c
c  author:  Hugh P. Summers, University of Strathclyde
c           JA7.08
```

```
c           Tel.: +44 (0)141-548-4196
c
c  date:    24 July 2002
c
c
c  version: 1.1   Hugh Summers  24/07/02
c  modified:      first release
c-----------------------------------------------------------------------
      INTEGER             IENDS,       IFORMS,      ISTDIM
      REAL*8              DX,          X
```

8.11 h4gspc: Subroutine h4gspc from library adas8xx

```
C
       subroutine h4gspc( istdim ,
     &                    xa     , n      ,
     &                    c1     , c2     , c3     , c4
     &                  )
c-----------------------------------------------------------------------
c
c  ***************** fortran77 subroutine: h4gspc.for *********************
c
c  purpose: generate precursors of spline coefficients suitable
c           for both forward and backward interpolation
c
c  calling program: various
c
c
c  input : (i*4)  istdim  = dimensionality for splining arrays
c
c  input : (r*8)  xa(i)   = set of knots
c  input : (i*4)  n       = number of knots  (n.le.istdim)
c
c  output: (r*8)  c1(i,j) = 1st spline coefficient precursor
c  output: (r*8)  c2(i,j) = 2nd spline coefficient precursor
c  output: (r*8)  c3(i,j) = 3rd spline coefficient precursor
c  output: (r*8)  c4(i,j) = 4th spline coefficient precursor
c
c  routines:
c          routine    source    brief description
c          ------------------------------------------------------------
c          h4spl3     adas
c          i4unit     adas      fetch unit number for output of messages
c
c  author:  Hugh P. Summers, University of Strathclyde
c           JA7.08
c           Tel.: +44 (0)141-548-4196
c
c  date:    24 July 2002
c
c
c  version: 1.1   Hugh Summers  24/07/02
c  modified:      first release
c
c-----------------------------------------------------------------------
      INTEGER             ISTDIM,      N
      REAL*8              C1(ISTDIM,ISTDIM-1),      C2(ISTDIM,ISTDIM-1)
      REAL*8              C3(ISTDIM,ISTDIM-1),      C4(ISTDIM,ISTDIM-1)
      REAL*8              XA(ISTDIM)
```

8.12 h4lnft: Subroutine h4lnft from library adas8xx

```
C
       subroutine h4lnft( istdim ,
     &                    x      , xsa   , y     , ysa   , ict
     &                  )
c-----------------------------------------------------------------------
c
c  ***************** fortran77 subroutine: h4lnft.for *********************
c
c  purpose:  perform linear interpolation
c
c
c  input : (r*8)  x     = required x-value
c  input : (r*8)  xsa(i)= x-values
c  input : (r*8)  ysa(i)= y-values
c  input : (i*4)  ict   = number of values
c
c  input : (r*8)  y     = returned y-value
c
c  author:  Hugh P. Summers, University of Strathclyde
c           JA7.08
c           Tel.: +44 (0)141-548-4196
c
c  date:    24 July 2002
c
c
c  version: 1.1   Hugh Summers  24/07/02
c  modified:      first release
c-----------------------------------------------------------------------
      INTEGER             ICT,         ISTDIM
      REAL*8              X,           XSA(ISTDIM), Y
      REAL*8              YSA(ISTDIM)
```

8.13 h4mxwl: Subroutine h4mxwl from library adas8xx

```
C
       subroutine h4mxwl( z0    , z      , zeff   ,
     &                    ixtyp , acoeff , s      ,
     &                    wi    , wj     , ei     , ej     ,
     &                    nx    , xa     , omga   ,
     &                    nt    , tea    , upsa
     &                   )
c-----------------------------------------------------------------------
c
c  ***************** fortran77 subroutine: h4mxwl.for *********************
c
c  purpose:  analyse electron impact collision strength data and convert
c        to Maxwell averaged collision strengths.
c
c  notes:    data is fitted with approximate forms to aid interpolation
c            depending on the transition type. these are
c                 1. dipole
c                 2. non-dipole
c                 3. spin change
c                 4. other
c
c  subroutine:
c
c  input : (r*8)  z0       = nuclear charge
c  input : (r*8)  z        = ion charge
c  input : (r*8)  zeff     = ion charge +1
c  input : (i*4)  ixtyp    = 1  dipole transition
c                          = 2  non-dipole transition
c                          = 3  spin change transition
c                          = 4  other
c  input : (r*8)  wi       = lower level statistical weight
c  input : (r*8)  wj       = upper level statistical weight
c  input : (r*8)  ei       = lower level energy (in selected units)
c  input : (r*8)  ej       = upper level energy
c  input : (r*8)  acoeff   = A-value (s-1) (dipole case only)
c  input : (i*4)  nx       = number of X-parm/omega pairs
c  input : (i*4)  nt       = number of output temperatures
c  input : (r*8)  xa(i)    = input X-parm. values
c  input : (r*8)  omga(i)  = input omega values
c  input : (r*8)  tea(i)   = output electron temps.(K)
c
c  output: (r*8)  upsa(i)  = output upsilons
c  output: (r*8)  s        = line strength (dipole case) otherwise zero

c          (i*4)  iibts  = 0  bad point option off
c                        = 1  bad point option on
c          (i*4)  iifpt  = 1  select one point optimising
c                        = 2  select two point optimising
c          (i*4)  iixop  = 0  optimising off
c                        = 1  optimising on  (if allowed)
c          (i*4)  iidif  = 0  ratio fitting for dipole x-sect
c                           (only with optimising)
c                        = 1  difference fitting for dipole x-sect
c
c  routines:
c          routine    source    brief description
c          ------------------------------------------------------------
c          egasm      adas
c          h4sort     adas     sorts a vector pair by the first vector
```

```
c          h4ftsp     adas     evaluate from a spline interpolant
c          h4gspc     adas     generate spline precursors
c          h4lnft     adas     perform linear interpolation
c          h4fasy     adas     evaluate  from spline. of specified asympts.
c          h4gasy     adas     create  specified asympts. for spline
c          h4form     adas     form for spline independent variable
c          h4spl3     adas     calculate spline of various end conditions
c          eei        adas     evaluates exp(x)*E1(x)
c          ee2        adas     evaluates exp(x)E2(x)
c          i4unit     adas     fetch unit number for output of messages
c
c  author:  h. p. summers, university of strathclyde
c           ja7.08
c           tel. 0141-548-4196
c
c  date:   19/07/02
c
c  update:
c
c-----------------------------------------------------------------------
      INTEGER             IXTYP,       NT,          NX
      REAL*8              ACOEFF,      EI,          EJ
      REAL*8              OMGA(NX),    S,           TEA(NT)
      REAL*8              UPSA(NT),    WI,          WJ,          XA(NX)
      REAL*8              Z,           Z0,          ZEFF
```

8.14 h4spl3: Subroutine h4spl3 from library adas8xx

```
C
       subroutine h4spl3( istdim ,
     &                    n      , h      , w
     &                  )
c-----------------------------------------------------------------------
c
c  ***************** fortran77 subroutine: h4spl3.for ******************
c
c  purpose: calculate splines with various end conditions
c
c  calling program: h4gspc
c
c
c  notes: (1) uses labelled common  /espl3/
c         (2) conditions at 1st & nth nodes controlled by iend1 & iendn
c           iend = 1 : specified d log(y) ie. dy/y at node stored in
c                      appropriate vector
c                = 2 : zero curvature
c                = 3 : constant curvature
c                = 4 : matched to specified functional form in terms of
c                      two parameters a and b such that
c                        funct = p(1)*a+q(1)*b
c                        1st deriv. = p(2)*a+q(2)*b
c                        2nd deriv. = p(3)*a+q(3)*b
c                        where a1,b1,p1,q1 are used for 1st node and
c                        an,bn,pn,qn for nth node
c
c  input : (i*4)  istdim  = dimensionality for splining arrays
c
c  input : (i*4)  n    = number of knots
c  input : (r*8)  h()  = intervals between knots
c
c  output: (r*8)  w(,) = spline matrix
c
c  routines:
c           routine    source    brief description
c           ------------------------------------------------------------
c           h4spl3     adas
c           i4unit     adas      fetch unit number for output of messages
c
c  author:  Hugh P. Summers, University of Strathclyde
c           JA7.08
c           Tel.: +44 (0)141-548-4196
c
c  date:    24 July 2002
c
c
c  version: 1.1   Hugh Summers  24/07/02
c  modified:      first release
c
c-----------------------------------------------------------------------
      INTEGER             ISTDIM,      N
      REAL*8              H(ISTDIM),   W(ISTDIM,ISTDIM)
```

8.15 h9gett: Subroutine h9gett from library adas8xx

```
       subroutine h9gett( iunit  , ndlev  ,
     &                    nv     , tva
     &                  )
c-----------------------------------------------------------------------
c
c  ****************** fortran77 subroutine: h9gett *********************
c
c  purpose:  to fetch temperature set from input adf04 type 3 data set.
c
c  calling program: adas809
c
c  data:
c           the 'real' data in the file is represented in an abbreviated
c           form which omits the "d" or "e" exponent specifier.
c           e.g. 1.23d-06 or 1.23e-06 is represented as 1.23-06
c                6.75d+07 or 6.75e+07 is represented as 6.75+07
c
c           therefore the form of each 'real' number in the data set is:
c                          n.nn+nn or n.nn-nn
c
c           the units used in the data file are taken as follows:
c
c           ionisation potential: wave number (cm-1)
c           index level energies: wave number (cm-1)
c           temperatures        : kelvin
c           a-values            : sec-1
c           gamma-values        :
c           rate coefft.        : cm3 sec-1
c
c
c  subroutine:
c
c  input : (i*4)  iunit   = unit to which input file is allocated
c  input : (i*4)  ndlev   = maximum number of levels that can be read
c
c  output: (i*4)  nv      = input data file: number of gamma/temperature
c                           pairs for a given transition.
c  output: (r*8)  scef()  = input data file: electron temperatures (k)
c
c          (i*4)  nvmax   = parameter = max. number of temperatures
c                                       that can be read in.
c
c          (i*4)  i4unit  = function (see routine section below)
c          (i*4)  iqs     = x-sect data format selector
c                           note: iqs=3 or 4 only allowed in this program
c          (i*4)  i       = general use.
c          (i*4)  j       = general use.
c          (i*4)  iline   = energy level index for current line
c          (i*4)  itpow() = temperatures - exponent
c                           note: mantissa initially kept in 'scef()'
c
c          (r*4)  zf      = should be equivalent to 'iz1'
c
c          (c*80) cline   = current energy level index parameter line
c          (c*500)buffer  = general string buffer storage
c          (c*3)  citpow()= used to parse values to itpow()
c          (c*5)  cscef() = used to parse values to scef()
c
c          (l*4)  ldata   = identifies  whether  the end of an  input
```

```
c                           section in the data set has been located.
c                           (.true. => end of section reached)
c          (l*4)  ltied() = .true.  => specified level tied
c                         = .false. => specified level is untied
c                           dimension => level index
c
c routines:
c          routine    source    brief description
c          ------------------------------------------------------------
c          i4unit     adas      fetch unit number for output of messages
c
c
c routines: none
c
c author:  Hugh Summers, University of strathclyde
c          ja7.08
c          ext. 4196
c
c date:    13/08/01
c
c update:
c
c unix-idl port:
c
c version: 1.1                          date: 14-06-013
c modified: Hugh Summers
c               - first version
c
c-----------------------------------------------------------------------
c-----------------------------------------------------------------------
      INTEGER             IUNIT,       NDLEV,       NV
      REAL*8              TVA(NVMAX)
```

8.16 h9int: Subroutine h9int from library adas8xx

```
      subroutine h9int( itran  , stcode  , i1a    , i2a   , aval  ,
     &                  iadftyp, DTYPE  , ADF37  , iz1   , MAXT  ,
     &                  beth   , il     , nv     , ia    , wa    ,
     &                  xja    , omga   , scx    , ilbeth, DPARAM,
     &                  zpla   , bwnoa  , ipla   , TINE  , IFOUT,
     &                  upsilon , dnsilon   )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: H9INT *********************
C
C  PURPOSE: GENERATES MAXWELLIAN AND NON-MAXWELLIAN UPSILONS AND
C           DOWNSILONS FROM GIVEN COLLISIONAL DATA. THIS PROGRAM
C           IS A SUBROUTINE VERSION OF ADAS809.
C
C  CALLING PROGRAM: CALLED FROM IDL VIA A C INTERFACE
C
C  SUBROUTINE:
C
C          (I*4)  NDLEV   = PARAMETER = MAX. NUMBER OF LEVELS ALLOWED
C          (I*4)  NDTRN   = PARAMETER = MAX. NO. OF TRANSITIONS ALLOWED
C          (I*4)  NEDIM   = PARAMETER = MAX. OF INPUT DATA FILE ENERGIES
C          (I*4)  NDTEM   = PARAMETER = MAXIMUM OF DATA CONVERSION TEMPS
C          (I*4)  NDTIN   = PARAMETER = MAX.NUMBER OF ISPF ENTERED TEMPS
C          (I*4)  NFDIM   = PARAMETER = MAX.NO. OF ENERGIES IN NUM. DIST.
C
C          (I*4)  I4UNIT  = FUNCTION (SEE ROUTINE SECTION BELOW)
C          (I*4)  IZ1     = RECOMBINING ION CHARGE READ FROM INPUT FILE
C                           (NOTE: IZ1 SHOULD EQUAL IZ+1)
C          (I*4)  IL      = INPUT DATA FILE: NUMBER OF ENERGY INDEX
C                                            LEVELS.
C          (I*4)  ITRAN   = INPUT DATA FILE: TOTAL NUMBER OF TRANSITIONS.
C          (I*4)  ICNTE   = NUMBER OF ELECTRON IMPACT TRANSITIONS INPUT
C          (I*4)  ICNTP   = NUMBER OF PROTON IMPACT TRANSITIONS INPUT
C          (I*4)  ICNTR   = NUMBER OF FREE ELECTRON RECOMBINATIONS INPUT
C          (I*4)  ICNTH   = NO. OF CHARGE EXCHANGE RECOMBINATIONS INPUT
C          (I*4)  NV      = INPUT DATA FILE: NUMBER OF GAMMA/TEMPERATURE
C                                       PAIRS FOR THE SELECTED TRANSITION.
C          (I*4)  MAXT    = NUMBER OF TEMPERATURE VALUES.
C          (I*4)  IFOUT   = 1 => 'TINE(ARRAY)' UNITS: KELVIN
C                         = 2 => 'TINE(ARRAY)' UNITS: EV
C                         = 3 => 'TINE(ARRAY)' UNITS:REDUCED TEMPERATURE
C          (I*4)  IUPPER  = SELECTED TRANSITION: UPPER LEVEL ARRAY INDEX
C          (I*4)  ILOWER  = SELECTED TRANSITION: LOWER LEVEL ARRAY INDEX
C          (I*4)  LUPPER  = SELECTED TRANSITION: UPPER INDEX LEVEL
C          (I*4)  LLOWER  = SELECTED TRANSITION: LOWER INDEX LEVEL
C
C          (R*8)  R8TCON  = FUNCTION (SEE ROUTINE SECTION BELOW)
C          (R*8)  EUPPER  = SELECTED TRANSITION: UPPER ENERGY LEVEL
C                           RELATIVE TO INDEX LEVEL 1. (CM-1)
C          (R*8)  ELOWER  = SELECTED TRANSITION: LOWER ENERGY LEVEL
C                           RELATIVE TO INDEX LEVEL 1. (CM-1)
C          (R*8)  WUPPER  = SELECTED TRANSITION: UPPER LEVEL STAT. WT.
C          (R*8)  WLOWER  = SELECTED TRANSITION: LOWER LEVEL STAR. WT.
C                           (NOTE: STAT. WT. = STATISTICAL WEIGHT)
C          (R*8)  BWNO    = IONISATION POTENTIAL (CM-1)
C          (R*8)  AA      = SELECTED TRANSITION A-VALUE (SEC-1)
C
C          (I*4)  IA()    = ENERGY LEVEL INDEX NUMBER
C          (I*4)  I1A()   = TRANSITION:
```

```
C                          LOWER ENERGY LEVEL INDEX (CASE ' ' & 'P')
C                          NOT USED                 (CASE 'H' & 'R')
C          (I*4)  I2A()   = TRANSITION:
C                           UPPER ENERGY LEVEL INDEX (CASE ' ' & 'P')
C                           CAPTURING    LEVEL INDEX (CASE 'H' & 'R')
C          (I*4)  IETRN() = ELECTRON IMPACT TRANSITION:
C                           INDEX VALUES IN MAIN TRANSITION ARRAYS WHICH
C                           REPRESENT ELECTRON IMPACT TRANSITIONS.
C          (I*4)  IPTRN() = PROTON IMPACT TRANSITION:
C                           INDEX VALUES IN MAIN TRANSITION ARRAYS WHICH
C                           REPRESENT PROTON IMPACT TRANSITIONS.
C          (I*4)  IRTRN() = FREE ELECTRON RECOMBINATION:
C                           INDEX VALUES IN MAIN TRANSITION ARRAYS WHICH
C                           REPRESENT FREE ELECTRON RECOMBINATIONS.
C          (I*4)  IHTRN() = CHARGE EXCHANGE RECOMBINATION:
C                           INDEX VALUES IN MAIN TRANSITION ARRAYS WHICH
C                           REPRESENT CHARGE EXCHANGE RECOMBINATIONS.
C          (I*4)  IE1A()  = ELECTRON IMPACT TRANSITION:
C                           LOWER ENERGY LEVEL INDEX
C          (I*4)  IE2A()  = ELECTRON IMPACT TRANSITION:
C                           UPPER ENERGY LEVEL INDEX
C          (I*4)  IP1A()  = PROTON IMPACT TRANSITION:
C                           LOWER ENERGY LEVEL INDEX
C          (I*4)  IP2A()  = PROTON IMPACT TRANSITION:
C                           UPPER ENERGY LEVEL INDEX
C
C          (R*8)  XJA()   = QUANTUM NUMBER (J-VALUE) FOR LEVEL 'IA()'
C                           NOTE: (2*XJA)+1 = STATISTICAL WEIGHT
C          (R*8)  WA()    = ENERGY RELATIVE TO LEVEL 1 (CM-1) FOR LEVEL
C                           'IA()'
C          (I*4)  IPLA(,) = PARENT INDEX FOR CONTRIBUTION TO IONIS.
C                           OF LEVEL
C                           1ST DIMENSION: PARENT INDEX
C                           2ND DIMENSION: LEVEL INDEX
C          (R*8)  ZPLA(,) = EFF. ZETA PARAM. FOR CONTRIBUTION TO IONIS.
C                           OF LEVEL
C                           1ST DIMENSION: PARENT INDEX
C                           2ND DIMENSION: LEVEL INDEX
C          (R*8)  AVALE() = ELECTRON IMPACT TRANSITION: A-VALUE (SEC-1)
C          (R*8)  AVAL()  = TRANSITION:
C                           A-VALUE (SEC-1)          (CASE ' ')
C                           NEUTRAL BEAM ENERGY      (CASE 'H')
C                           NOT USED                 (CASE 'P' & 'R')C
C          (R*8)  SCX()   = INPUT DATA FILE: X-PARAMETER SET
C          (R*8)  SCEF()  = PROGRAM: INPUT FILE CONVERSION TEMPS, OR 2/3
C                           AVERAGE ENERGY OF NUMERICAL DISTRIBUTION IF
C                           ADF37 COMPARISON FILE IS SELECTED (KELVIN)
C          (R*8)  SCEF2() = MOST COMMON ENERGY OF DISTRIBUTION FUNC. IF
C                           ADF37 COMPARISON FILE IS SELECTED (KELVIN)
C          (R*8)  GAMMA() = INPUT DATA FILE: SELECTED TRANSITION -
C                                            GAMMA VALUE AT 'SCEF()'
C          (R*8)  RATE()  = INPUT DATA FILE: SELECTED TRANSITION -
C                           EXCITATION RATE COEFF. (CM**3/S) AT 'SCEF()'
C          (R*8)  DRATE() = INPUT DATA FILE: SELECTED TRANSITION -
C                           DEEXCITATION RATE COEF.(CM**3/S) AT 'SCEF()'
C
C          (R*8)  TINE()  = ISPF ENTERED ELECTRON TEMPERATURE VALUES.
C                           (NOTE: UNITS ARE GIVEN BY 'IFOUT')
C          (R*8)  TOA()   = ISPF ENTERED TEMPERATURES (KELVIN)
C
C          (R*8)  TOSA()  = SPLINE: SELECTED TEMPERATURES (KELVIN)
```

```
C          (R*8)  GAMOSA()= SPLINE INTEROPLATED GAMMA VALUE AT 'TOSA()'
C          (R*8)  ROSA()  = EXCITATION RATE COEFF.(CM**3/S) AT 'TOSA()'
C          (R*8)  DROSA() = DEEXCITATION RATE COEF.(CM**3/S) AT 'TOSA()'
C
C          (R*8)  TOMA()  = MINIMAX: SELECTED TEMPERATURES (KELVIN)
C          (R*8)  GAMOMA()= MINIMAX GENERATED GAMMA VALUE AT 'TOMA()'
C          (R*8)  ROMA()  = EXCITATION RATE COEFF.(CM**3/S) AT 'TOMA()'
C          (R*8)  DROMA() = DEEXCITATION RATE COEF.(CM**3/S) AT 'TOMA()'
C
C          (R*8)  TSCEF(,) = INPUT DATA FILE: ELECTRON TEMPS OR 2/3
C                            AVERAGE ENERGY IF ADF37 COMPARISON FILE IS
C                            SELECTED
C                            1ST DIMENSION: TEMPERATURES (SEE 'SCEF()')
C                            2ND DIMENSION: 1 => KELVIN  (IFOUT=1)
C                                           2 => EV      (IFOUT=2)
C                                           3 => REDUCED (IFOUT=3)
C          (R*8)  SCOM(,) = TRANSITION:
C                            GAMMA VALUES             (CASE ' ' & 'P')
C                            RATE COEFFT. (CM3 SEC-1) (CASE 'H' & 'R')
C                           1ST DIMENSION - TEMPERATURE 'SCEF()'
C                           2ND DIMENSION - TRANSITION NUMBER
C
C          (L*4)  LTRNG() = .TRUE.  => TEMPERATURE 'TOA()' IN RANGE
C                           .FALSE. => TEMPERATURE 'TOA()' OUT OF RANGE
C                           (RANGE = INPUT TEMPERATURE RANGE)
C
C          (C*1)  TCODE() = TRANSITION: DATA TYPE POINTER:
C                           ' ' => ELECTRON IMPACT   TRANSITION
C                           'P' => PROTON   IMPACT   TRANSITION
C                           'H' => CHARGE   EXCHANGE RECOMBINATION
C                           'R' => FREE     ELECTRON RECOMBINATION
C          (C*18) CSTRGA()= NOMENCLATURE/CONFIGURATION FOR LEVEL 'IA()'
C          (C*18) CSTRGB()= AS CSTRGA() BUT ONLY TAKING THE LAST
C                           'ICSTMX' NON-BLANK BYTES.
C          (C*22) STRGA() = NOMENCLATURE FOR LEVEL 'IA()' INCLUDES:
C                           'CSTRGA()' AND QUANTUM NUMBERS.
C    (I*4)  IFIRST  = FIRST NON-BLANK CHARCTER IN FILENAME
C
C    (I*4)  ILAST   = LAST  NON-BLANK CHARCTER IN FILENAME
C
C          (L*4)  OPEN07  = .TRUE.  => UNIT 7 IS OPEN
C                           .FALSE.  => UNIT 7 IS CLOSED
C          (L*4)  LREP    = .TRUE.  => PAPER.TXT TO BE REPLACED
C                           .FALSE.  => PAPER.TXT NOT TO BE REPLACED
C
C
C NOTE:    INPUT TEMPERATURES 'TINE()' ARE CONVERTED TO KELVIN 'TOA()'.
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C
C  AUTHOR:  PAUL BRYANS
C
C  DATE:    02 MARCH 2005
C
C  MODIFICATION HISTORY:
C
C  DATE:   02/03/05 VERSION: 1.1
C  MODIFIED: PAUL BRYANS
C - EXAMPLE CODE
```

```
C
C  DATE:   17/03/05 VERSION: 1.2
C  MODIFIED: ALLAN WHITEFORD
C - MODIFIED TO INTERFACE WITH C/IDL.
C  DATE:   04/04/05
C  MODIFIED: ALLAN WHITEFORD
C - CHANGED LSS04A FROM A SCALAR TO AN ARRAY OF
C                 SIZE (NDLEV,NDMET)
C
C-----------------------------------------------------------------------
      CHARACTER*80        ADF37
      INTEGER             DTYPE,       I1A(NDTRN),  I2A(NDTRN)
      INTEGER             IA(NDLEV),   IADFTYP,     IFOUT,       IL
      INTEGER             ILBETH,      IPLA(NDMET,NDLEV),        ITRAN
      INTEGER             IZ1,         MAXT,        NV
      INTEGER             STCODE(NDTRN)
      REAL*8              AVAL(NDTRN), BETH(NDTRN), BWNOA(NDMET)
      REAL*8              DNSILON(NDTEM,NDTRN),     DPARAM
      REAL*8              OMGA(NEDIM,NDTRN),        SCX(NEDIM)
      REAL*8              TINE(NDTIN), UPSILON(NDTEM,NDTRN)
      REAL*8              WA(NDLEV),   XJA(NDLEV),  ZPLA(NDMET,NDLEV)
```

8.17 h9ispf: Subroutine h9ispf from library adas8xx

```
      SUBROUTINE H9ISPF( LPEND    , NDTIN    ,
     &                   IL       , NDTEM    ,
     &                   NV       , TSCEF    , TSCEF2   ,
     &                   ITRAN    , I1A      , I2A      ,
     &                   FTYPE    , TITLE    ,
     &                   ISTRN    , IFOUT    ,
     &                   MAXT     , TINE     ,
     &                   LFSEL    , LOSEL    ,
     &                   TOLVAL   ,
     &                   DTYPE    , KAP_VAL  , DRU_VAL  ,
     &                   ADF37
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: H9ISPF *********************
C
C  PURPOSE: PIPE COMMUNICATIONS WITH IDL AND TO RETURN USER SELECTED
C           OPTIONS AND VALUES.
C
C  CALLING PROGRAM: ADAS809
C
C  SUBROUTINE:
C
C  I/O   : (L*4)   LPEND    = .TRUE.  => END ANALYSIS OF CURRENT DATA
C                                        SETS
C                            = .FALSE. => CONTINUE PANALYSIS WITH CURRENT
C                                        DATA SETS
C
C  INPUT : (I*4)   NDTIN    = MAX. NUMBER OF TEMPERATURES ALLOWED
C
C  INPUT : (I*4)   IL       = NUMBER OF ENERGY LEVELS
C
C  INPUT : (I*4)   NDTEM    = INPUT DATA FILE: MAX. NO. OF TEMPERATURES
C
C  INPUT : (I*4)   NV       = INPUT DATA FILE: NUMBER   OF TEMPERATURES
C
C  INPUT : (R*8)   TSCEF(,) = INPUT DATA FILE: ELECTRON TEMPERATURES, OR
C                             2/3 AVERAGE ENERGY OF A NUMERICAL
C                             DISTRIBUTION
C                             2ND DIMENSION: 1 => KELVIN  (IFOUT=1)
C                                            2 => EV      (IFOUT=2)
C                                            3 => REDUCED (IFOUT=3)
C
C  INPUT : (R*8)   TSCEF2(,)= INPUT DATA FILE: MOST COMMON ELECTRON
C                            'TEMPERATURE' FOR A NUMERICAL DISTRIBUTION,
C                             OR A ZERO ARRAY IN ANALYTIC CASE
C                             2ND DIMENSION: 1 => KELVIN  (IFOUT=1)
C                                            2 => EV      (IFOUT=2)
C                                            3 => REDUCED (IFOUT=3)
C
C  INPUT : (I*4)   ITRAN    = NUMBER OF ELECTRON IMPACT TRANSITIONS
C
C  INPUT : (I*4)   I1A()    = ELECTRON IMPACT TRANSITION:
C                               LOWER ENERGY LEVEL INDEX
C
C  INPUT : (I*4)   I2A()    = ELECTRON IMPACT TRANSITION:
C                               UPPER ENERGY LEVEL INDEX
C
C  INPUT : (C*6)   FTYPE    = TYPE OF FILE SELECTED AS COMPARISON
```

```
C
C  OUTPUT: (C*40)  TITLE    = USER ENTERED GENERAL TITLE FOR PROGRAM RUN
C
C  OUTPUT: (I*4)   ISTRN    = SELECTED ELECTRON IMPACT TRANSITION INDEX
C
C  OUTPUT: (I*4)   IFOUT    = 1 => INPUT TEMPERATURES IN KELVIN
C                           = 2 => INPUT TEMPERATURES IN EV
C                           = 3 => INPUT TEMPERATURES IN REDUCED FORM
C
C  OUTPUT: (I*4)   MAXT     = NUMBER OF INPUT TEMPERATURES ( 1 -> 20)
C
C  OUTPUT: (R*8)   TINE()   = ELECTRON TEMPERATURES (UNITS: SEE 'IFOUT')
C
C  OUTPUT: (L*4)   LFSEL    = .TRUE.  => CARRY OUT MINIMAX POLYNOMIAL
C                                        FITTING
C                           = .FALSE. => - DO NOT DO THE ABOVE -
C
C  OUTPUT: (R*8)   TOLVAL   = FRACTIONAL TOLERANCE FOR MINIMAX FIT
C                             (=0 IF MINIMAX FIT NOT SELECTED)
C
C  OUTPUT: (L*4)   LOSEL    = .TRUE.  => CALCULATE INTERPOLATED VALUES
C                                        FOR OUTPUT.
C                           = .FALSE. => - DO NOT DO THE ABOVE -
C
C  OUTPUT: (I*8)   DTYPE    = DISTRIBUTION TYPE SUCH THAT:
C                             0 => MAXWELLIAN
C                             1 => KAPPA
C                             2 => NUMERICAL
C  OUTPUT: (R*8)   KAP_VAL  = VALUE OF KAPPA
C
C
C
C     (I*4)   ILOGIC   = RETURN VALUE FROM IDL WHICH IS USED TO
C       REPRESENT A LOGICAL VARIABLE SINCE IDL
C                          DOES HAVE SUCH DATA TYPES.
C
C    (I*4)   I     = GENERAL PURPOSE COUNTER
C
C    (I*4)   J     = GENERAL PURPOSE COUNTER
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C    XXFLSH     IDL-ADAS  CALLS 'FLUSH' TO CLEAR PIPE
C
C AUTHOR:  HUGH SUMMERS (UNIV. OF STRATHCLYDE)
C
C DATE:    30/11/01
C
C DATE:   07/04/05
C MODIFIED: ALLAN WHITEFORD
C - ADDED FORMAT STATEMENT TO READING FROM PIPE
C-----------------------------------------------------------------------
      CHARACTER*80        ADF37
      CHARACTER*6         FTYPE
      CHARACTER*40        TITLE
      INTEGER             DTYPE,       I1A(ITRAN),  I2A(ITRAN),  IFOUT
      INTEGER             IL,          ISTRN,       ITRAN,       MAXT
      INTEGER             NDTEM,       NDTIN,       NV
      LOGICAL             LFSEL,       LOSEL,       LPEND
      REAL*8              DRU_VAL,     KAP_VAL,     TINE(NDTIN), TOLVAL
      REAL*8              TSCEF(NDTEM,3),           TSCEF2(NDTEM,3)
```

8.18 h9ntqd: Subroutine h9ntqd from library adas8xx

```
       subroutine h9ntqd  ( nedim   , ntdim   , nfdim  ,
     &                      ifint   , itype   , itypt  , ilinr , iescl ,
     &                      itrn    ,
     &                      ne      , nt      ,
     &                      evt     , xa      ,
     &                      oma     , tva     ,
     &                      upsilon , dnsilon ,
     &                      kappa   , dru_val , dist   ,
     &                      nef     , en      , f      ,
     &                      ntf     , tvf     ,
     &                      lbeth   , beth    ,
     &                      nform1  , param1  , nform2 , param2
     &                     )
C-----------------------------------------------------------------------
C
C  VERSION:   1.0
C
C  PURPOSE:  Executes quadratures over collision strengths to form
C            excitation and de-excitation effective collision strengths
C            for atoms and ions with tabulated collision stregths
C            as a function of x parameter.
C
C            Quadrature can be executed over a Maxwellian, kappa
C            distribution, Druyvesteyn or numerical distribution.
C            Linear interpoation is recomended and is default (ilinr=1)
C            Quadratic interpolation is also allowed for analytic
C            distributions.
C            1/E varaible interpolation is allowed for Maxwellian only.
C
C  DATA:
C
C  PROGRAM:
C
C          (i*4)  ifint   = input     = indep. var. for interpolation
C             (1 = E)
C             (2 = 1/E)
C          (i*4)  itype   = input     = collision strength type, to give
C high energy behaviour
C             (1 = dipole --> a*log(X+b) )
C             (2 = non-dp --> a+X/b      )
C             (3 = spin ch--> a/(X+b)**2 )
C          (i*4)  itypt   = input     = threshold behaviours allowed
C             (1(ion)     = const to 1st pt.)
C             (2(neutral) = 0     to 1st pt.)
C          (i*4)  ilinr   = input     = allow linear or quadratic interp
C             (1 = linear   )
C             (2 = quadratic)
C          (i*4)  iescl   = input     = allow e**2*omega + lin. interp
C             (1 = normal use)
C             (2 = e**2*omega +lin.)
C iescl=2 not implimented
C          (i*4)  itrn    = input     = index of current transition
C          (i*4)  nedim   = input     = max no of energies in omega file
C          (i*4)  ntdim   = input     = max no of temperatures
C          (i*4)  nfdim   = input     = max no of energies in adf37 file
C          (i*4)  ne   = input     = number of energies in omega file
C          (i*4)  nef   = input     = number of energies in adf37 file
C          (i*4)  nt   = input     = number of temperatures
C          (i*4)  ntf   = input     = number of temperatures in adf37
```

```
C          (r*8)  xa()    = input     = tabul. x param. for coll. str.
C          (r*8)  evt   = input     = theshold energy (eV)
C          (r*8)  oma()   = input     = tabul. coll. str.
C          (r*8)  tva()   = input     = temperatures (eV)
C          (r*8)  tvf()   = input     = temperatures (eV) from adf37
C          (i*4)  dist    = input     = electron distribution
C             (0 = Maxwellian )
C             (1 = kappa    )
C             (2 = numerical  )
C             (3 = Druyvesteyn)
C          (r*8)  kappa   = input     = kappa value of electron dist.
C          (r*8)  dru_val = input     = x parameter from Druyvesteyn dist.
C          (r*8)  en(,)   = input     = adf37 energy (eV)
C          (r*8)  f(,)    = input     = adf37 distribution
C          (l*4)  lbeth   = input     = true if limit point exists
C          (r*8)  beth    = input     = infinite energy limit point of omega
C          (i*4)  nform1  = input     = type of threshold behaviour
C            (1 = cutoff     )
C            (2 = energy^param1)
C          (r*8)  param1  = input     = parameter of threshold form
C          (i*4)  nform2  = input     = type of high-energy behaviour
C            (1 => cutoff         )
C            (2 => energy^-param2(1)     )
C            (3 => exp(-param2(1)*energy))
C          (r*8)  param2()= input     = parameter of high-energy form
C
C          (r*8)  xf    = program   = current en(i)/evt
C          (r*8)  omega(,)= program   = oma interpolated to distribution
C          function energy grid
C          (r*8)  sumi()  = program   = gamma contrib. from i  -> i+1
C          (r*8)  sumn()  = program   = gamma contrib. from ne-1  -> ne
C          (r*8)  sumu()  = program   = gamma contrib. from ne --> inf.
C          (r*8)  suml()  = program   = gamma contrib. from thres. -> 1
C          (r*8)  en()    = program   = tabul. ener. for coll. str. (ev)
C          (r*8)  fva()   = program   = indep. var. for interpolation
C          (r*8)  expi()  = program   = current exp(-(ui-ut))
C          (r*8)  expi1() = program   = current exp(-(ui1-ut))
C          (r*8)  exp1()  = program   = exp(-(u1-ut))
C          (r*8)  ui()    = program   = current eva(i)/kte
C          (r*8)  ui1()   = program   = current eva(i+1)/kte
C          (r*8)  u1()    = program   = eva(1)/kte
C          (r*8)  ut    = program   = evt/kte
C          (r*8)  uj()    = program   = ui-ut
C          (r*8)  uj1()   = program   = ui1-ut
C          (r*8)  w0    = program   = interpolation working variable
C          (r*8)  w1    = program   = interpolation working variable
C          (r*8)  w2    = program   = interpolation working variable
C          (r*8)  v0    = program   = interpolation working variable
C          (r*8)  v1    = program   = interpolation working variable
C          (r*8)  v2    = program   = interpolation working variable
C          (r*8)  y1    = program   = interpolation working variable
C          (r*8)  y2    = program   = interpolation working variable
C          (r*8)  c0    = program   = interpolation working variable
C          (r*8)  c1    = program   = interpolation working variable
C          (r*8)  c2    = program   = interpolation working variable
C          (r*8)  cc0    = program   = interpolation working variable
C          (r*8)  cc1    = program   = interpolation working variable
C          (r*8)  ww0    = program   = interpolation working variable
C          (r*8)  ww1    = program   = interpolation working variable
C          (r*8)  ww2    = program   = interpolation working variable
C          (r*8)  a1    = program   = interpolation working variable
```

```
C          (r*8)  a2   = program   = interpolation working variable
C          (r*8)  b1   = program   = interpolation working variable
C          (r*8)  b2   = program   = interpolation working variable
C
C          (r*8)  upsilon(,)  = output    = upsilon values
C          (r*8)  dnsilon(,)  = output    = downsilon values
C
C  routines:
C     routine    source brief description
C     ------------------------------------------------------------
C     eei        copase evaluates exp(x)*E1(x)
C     ee2        copase evaluates exp(x)*E2(x)
C     lngama evaluates ln(gamma(a))
c     ingama evaluates incomplete gamma P(a,x)
c     ingamq evaluates incomplete gamma 1-P(a,x)
C
C author:  H P Summers
C     K1/1/57
C     JET ext. 4941
C
C date:    26/05/93
C
C update:  30/11/01  HP Summers - altered input to use x parameter
C
C update:  23/11/04  P Bryans - altered to evaluate non-maxwellian
C  electron distributions
C
C update:  20/07/07  A Whiteford - Modified comments slightly to allow
C                                  for automatic generation of
C                                  documentation.
C
C-----------------------------------------------------------------------
      INTEGER             DIST,         IESCL,       IFINT,       ILINR
      INTEGER             ITRN,         ITYPE,       ITYPT,       NE
      INTEGER             NEDIM,        NEF,         NFDIM,       NFORM1
      INTEGER             NFORM2,       NT,          NTDIM,       NTF
      LOGICAL             LBETH
      REAL*8              BETH,         DNSILON(NTDIM),           DRU_VAL
      REAL*8              EN(NTDIM,NFDIM),           EVT
      REAL*8              F(NTDIM,NFDIM),            KAPPA
      REAL*8              OMA(NEDIM),   PARAM1,      PARAM2(2)
      REAL*8              TVA(NTDIM),   TVF(NTDIM),  UPSILON(NTDIM)
      REAL*8              XA(NEDIM)
```

8.19 h9qd3b: Subroutine h9qd3b from library adas8xx

```
      subroutine h9qd3b( ntdim , nedim   , nfdim   ,
     &                   teff  , tenum   ,
     &                   ne    , nt      , wupper  , wlower  ,
     &                   xa    , dist    , kap_val , dru_val ,
     &                   zeta  , evt     , omega   ,
     &                   n_en  , en      , f       , nte     ,
     &                   nform1, param1  , nform2  , param2  ,
     &                   alpha , q
     &                 )

C-----------------------------------------------------------------------
C
C  ****************** fortran77 subroutine: h9qd3b *********************
C
C  purpose:  To execute quadratures over ionisation collision strengths
C            to determine the ionisation and 3-body recombination
C            coefficients. Free electron distribution function may be
C            Maxwellian, Kappa, Druyvesteyn, or numeric from adf37 file.
C
C  calling program: adas809
C
C  input : (i*4)  ntdim    = max no of temperatures that can be read in
C  input : (i*4)  nedim    = max no of energy points that can be read in
C  input : (r*8)  teff()   = input effective temperatures (eV)
C  input : (r*8)  tenum()  = effective temperatures from adf37 file (eV)
C  input : (i*4)  ne       = no of x parameter values in adf04 type 1
C  input : (i*4)  nt       = no of input temperatures
C  input : (r*8)  wupper   = statistical weight of upper level
C  input : (r*8)  wlower   = statistical weight of lower level
C  input : (r*8)  xa()     = x parameter from adf04 type1
C  input : (i*4)  dist     = distribution type
C                               0 => Maxwellian
C                               1 => kappa
C                               2 => numeric
C                               3 => Druyvesteyn
C  input : (r*8)  kap_val  = value of kappa parameter
C  input : (r*8)  dru_val  = value of x parameter in Druyvesteyn dist
C  input : (r*8)  evt      = ionisation potential
C  input : (r*8)  omega()  = collision strength from adf04 type1
C  input : (i*4)  n_en     = no of energy points in adf37
C  input : (r*8)  en(,)    = energy points of distribution tabulation
C  input : (r*8)  f(,)     = distribution function tabulation
C  input : (i*4)  nte      = no of temperatures in adf37
C  input : (i*4)  nform1   = type of threshold behaviour
C                               1 => cutoff
C                               2 => energy^param1
C  input : (r*8)  param1() = parameter of threshold form
C  input : (i*4)  nform2   = type of high-energy behaviour
C                               1 => cutoff
C                               2 => energy^-param2(1)
C                               3 => exp(-param2(1)*energy)
C  input : (r*8)  param2   = parameter of high-energy form
C
C  output: (r*8)  alpha    = 3-body recombination coefficient
C  output: (r*8)  q        = ionisation coefficient
C
C  routines:
C          routine    source    brief description
C          ------------------------------------------------------------
```

```
C          eei        ADAS      evaluates  exp(x)*E1(x)
C          lngama     ADAS evaluates ln(gamma(x))
C
C  author: Paul Bryans, University of Strathclyde
C
C  date:   22/11/04
C
c  modification history:
c
c  date:   22/11/04 version: 1.1
c  modified: Paul Bryans
c - first release
c
c  date:   26/11/04
c  modified: Allan Whiteford
c - made some arrays locally dimensioned to prevent
c                g77 compiled code failing when it tried to make
c                large automatic arrays.
c
c  date:   26/11/04
c  modified: Paul Bryans
c - Moved final exponential coefficient of Maxwellian
c                and Druyvesteyn ionisation rate inside integrals to
c                eliminate overflow problem when this becomes large.
c
c  date:   18/08/05
c  modified: Paul Bryans
c - Set energy difference equal to zero at threshold.
c                Numerical precision was causing this to be small and
c                negative, giving NaN when raised to non-integer power.
c
C-----------------------------------------------------------------------
      INTEGER             DIST,         NE,           NEDIM,       NFDIM
      INTEGER             NFORM1,       NFORM2,       NT,          NTDIM
      INTEGER             NTE,          N_EN
      REAL*8              ALPHA(NTDIM),               DRU_VAL
      REAL*8              EN(NTDIM,NFDIM),            EVT
      REAL*8              F(NTDIM,NFDIM),             KAP_VAL
      REAL*8              OMEGA(NEDIM),               PARAM1
      REAL*8              PARAM2(2),    Q(NTDIM),     TEFF(NTDIM)
      REAL*8              TENUM(QDNT),  WLOWER,       WUPPER
      REAL*8              XA(NEDIM),    ZETA
```

8.20 h9rate: Subroutine h9rate from library adas8xx

```
      SUBROUTINE H9RATE( nmax   ,  NARR  ,
     &                   TEMP   , GAMAUP , gamadn ,
     &                   EUPPER , ELOWER ,
     &                   WUPPER , WLOWER ,
     &                   RATE   , DRATE
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: H9RATE *********************
C
C  PURPOSE: TO CALCULATE THE EXCITATION AND DE-EXCIATATION RATE COEFFI-
C           CIENTS FOR A SET OF INPUT TEMPERATURE(kelvin)/ GAMMA PAIRS.
C
C  CALLING PROGRAM:  ADAS809
C
C  SUBROUTINE:
C
C  INPUT :  (I*4)  NMAX    = MAX NUMBER OF INPUT TEMPERATURE/GAMMA PAIRS
C  INPUT :  (I*4)  NARR    = NUMBER OF INPUT TEMPERATURE/GAMMA PAIRS
C  INPUT :  (R*8)  TEMP()  = TEMPERATURE VALUES (kelvin)
C  INPUT :  (R*8)  GAMUP() = GAMMA VALUES FOR EXCITATION
C  INPUT :  (R*8)  GAMDN() = GAMMA VALUES FOR DE-EXCITATION
C
C  INPUT :  (R*8)  EUPPER  = SELECTED TRANSITION - UPPER ENERGY LEVEL
C                            RELATIVE TO INDEX LEVEL 1 (CM-1).
C  INPUT :  (R*8)  ELOWER  = SELECTED TRANSITION - LOWER ENERGY LEVEL
C                            RELATIVE TO INDEX LEVEL 1 (CM-1).
C
C  INPUT :  (R*8)  WUPPER  = SELECTED TRANSITION - UPPER ENERGY LEVEL
C                            STATISTICAL WEIGHT.
C  INPUT :  (R*8)  WLOWER  = SELECTED TRANSITION - LOWER ENERGY LEVEL
C                            STATISTICAL WEIGHT.
C
C  OUTPUT:  (R*8)  RATE    = EXCITATION RATE COEFFS (cm**3/s)
C  OUTPUT:  (R*8)  DRATE   = DEEXCITATION RATE COEFS (cm**3/s)
C
C           (R*8)  TK2ATE  = PARAMETER = EQUATION CONSTANT = 1.5789D+05
C           (R*8)  R2GAM   = PARAMETER = EQUATION CONSTANT = 2.17161D-08
C           (R*8)  WN2RYD  = PARAMETER =
C                            WAVE NUMBER (CM-1) TO RYDBERG CONVERSION
C
C           (I*4)  I       = GENERAL ARRAY INDEX
C
C           (R*8)  SUPPER  = 1/(UPPER LEVEL STATISTICAL WEIGHT)
C           (R*8)  SLOWER  = 1/(LOWER LEVEL STATISTICAL WEIGHT)
C           (R*8)  RYDDIF  = NEGATIVE TRANSITION ENERGY IN RYDBERGS
C                            ( NOTE: 1 Rydberg = 1.09737E5 cm-1)
C           (R*8)  ATE     = EQUATION PARAMETER
C           (R*8)  GVAL    = EQUATION PARAMETER
C
C ROUTINES:  NONE
C
C NOTES:
C            EQUATIONS USED -
C
C                      2.17161E-8 x GAMMA x SQRT(157890 / TEMP)
C            RATE = ------------------------------------------
C                   WLOWER x EXP(1.4388 x (EUPPER-ELOWER) / TEMP)
C
```

```
C                      2.17161E-8 x GAMMA x SQRT(157890 / TEMP)
C            DRATE = -------------------------------------------
C                                    WUPPER
C
C AUTHOR:  HUGH SUMMERS (UNIVERSITY OF STRATHCLYDE)
C          JA7.08
C          EXT. 4196
C
C DATE:    30/11/01
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             NARR,         NMAX
      REAL*8              DRATE(NMAX), ELOWER,      EUPPER
      REAL*8              GAMADN(NMAX),             GAMAUP(NMAX)
      REAL*8              RATE(NMAX),  TEMP(NMAX),  WLOWER,      WUPPER
```

8.21 h9spln: Subroutine h9spln from library adas8xx

```
      SUBROUTINE H9SPLN( NDTRN   , NDTEM  ,
     &                   NDTIN   ,
     &                   NV      , MAXT    , NPSPL  ,
     &                   IETRN   , ICNTE   , DTYPE  ,
     &                   ISTRN   , ICNTS   ,
     &                   USCEF   , TOA     , TOSA   ,
     &                   SCOM    ,
     &                   GAMMA   , GAMOSA  ,
     &                   LTRNG   , ITRN
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: H9SPLN *********************
C
C  PURPOSE:
C         1) PERFORMS CUBIC SPLINE ON LOG(TEMPERATURE) VERSUS LOG(GAMMA)
C            INPUT DATA. ('SCEF' VERSUS 'GAMMA' , NV DATA PAIRS)
C
C         2) INTERPOLATES 'MAXT'  GAMMA VALUES USING ABOVE SPLINES AT
C            TEMPERATURES READ IN FROM ISPF PANELS FOR TABULAR OUTPUT.
C            (ANY TEMPERATURE VALUES WHICH REQUIRED EXTRAPOLATION TO
C             TAKE PLACE ARE SET TO ZERO).
C
C         3) INTERPOLATES 'NPSPL' GAMMA VALUES USING ABOVE SPLINES AT
C            TEMPERATURES EQUI-DISTANCE ON RANGE OF LOG(TEMPERATURES)
C            STORED IN INPUT 'SCEF' ARRAY.
C
C  CALLING PROGRAM: ADAS809
C
C
C  SUBROUTINE:
C
C
C  INPUT : (I*4)  NV      = INPUT DATA FILE: NUMBER OF GAMMA/TEMPERATURE
C                           PAIRS READ FOR THE TRANSITION BEING ASSESSED
C  INPUT : (I*4)  MAXT    = NUMBER OF ISPF ENTERED TEMPERATURE VALUES AT
C                           WHICH INTERPOLATED GAMMA VALUES ARE REQUIRED
C                           FOR TABULAR OUTPUT.
C  INPUT : (I*4)  NPSPL   = NUMBER OF  SPLINE  INTERPOLATED  GAMMA/TEMP.
C                           REQUIRED FOR GRAPHICAL DISPLAY.
C
C  INPUT : (I*4)  SCEF()  = INPUT DATA FILE: TEMPERATURES (KELVIN)
C  INPUT : (I*4)  TOA()   = ISPF PANEL ENTERED TEMPERATURES (KELVIN)
C  OUTPUT: (I*4)  TOSA()  = 'NPSPL' TEMPERATURES FOR GRAPHICAL OUTPUT
C                           (KELVIN).
C
C  INPUT : (R*8)  GAMMA() = INPUT DATA FILE: SELECTED TRANSITION -
C                           GAMMA VALUES AT 'SCEF()'.
C  OUTPUT: (R*8)  GAMOSA()= SPLINE INTERPOLATED GAMMA VALUES AT 'TOSA()'
C
C  OUTPUT: (L*4)  LTRNG() = .TRUE.  => OUTPUT   SPLINE    VALUE    WAS
C                                      INTERPOLATED FOR 'DLOG(TOA())'.
C                           .FALSE. => OUTPUT   SPLINE    VALUE    WAS
C                                      EXTRAPOLATED FOR 'DLOG(TOA())'.
C                                      (NOTE: 'YOUT()=0' AS 'IOPT < 0').
C
C          (I*4)  NIN     = PARAMETER = MAX. NO. OF  INPUT  TEMP/GAMMA
C                                      PAIRS MUST BE >= 'NV'
C          (I*4)  NOUT    = PARAMETER = MAX. NO. OF 'OUTPUT TEMP/GAMMA
```

```
C                                     PAIRS MUST BE >= 'MAXT' & 'NPSPL'
C
C          (I*4)  IARR    = ARRAY SUBSCRIPT USED FOR TEMP/GAMMA PAIRS
C          (I*4)  IOPT    = DEFINES THE BOUNDARY DERIVATIVES FOR THE
C                           SPLINE ROUTINE 'XXSPLE', SEE 'XXSPLE'.
C                           (VALID VALUES = <0, 0, 1, 2, 3, 4)
C
C          (R*8)  TSTEP   = THE SIZE OF STEP BETWEEN 'XOUT()' VALUES FOR
C                           GRAPHICAL  OUTPUT  TEMP/GAMMA  PAIRS  TO  BE
C                           CALCULATED USING SPLINES.
C
C          (L*4)  LSETX   = .TRUE.  => SET UP SPLINE PARAMETERS RELATING
C                                      TO 'XIN' AXIS.
C                           .FALSE. => DO NOT SET UP SPLINE PARAMETERS
C                                      RELATING TO 'XIN' AXIS.
C                                      (I.E. THEY WERE SET IN A PREVIOUS
C                                            CALL )
C                           (VALUE SET TO .FALSE. BY 'XXSPLE')
C
C          (R*8)  XIN()   = LOG( 'SCEF()' )
C          (R*8)  YIN()   = LOG( 'GAMMA()' )
C          (R*8)  XOUT()  = LOG(TEMPERATURES AT WHICH SPLINES REQUIRED)
C          (R*8)  YOUT()  = LOG(OUTPUT SPLINE INTERPOLATED GAMMA VALUES)
C          (R*8)  DF()    = SPLINE INTERPOLATED DERIVATIVES
C
C          (L*4)  LDUMP() = .TRUE.  => OUTPUT SPLINE VALUE INTRPOLATED
C                                      FOR 'YOUT()'.
C                           .FALSE. => OUTPUT SPLINE VALUE EXTRAPOLATED
C                                      FOR 'YOUT()'.
C                                      (NOTE: USED AS A DUMMY ARGUMENT.
C                                             ALL VALUES WILL BE TRUE.)
C
C NOTE:
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          XXSPLE     ADAS      SPLINE SUBROUTINE (EXTENDED DIAGNOSTICS)
C          R8FUN1     ADAS      REAL*8 FUNCTION: ( X -> X )
C
C AUTHOR:  PAUL BRYANS (UNIVERSITY OF STRATHCLYDE)
C
C DATE:    09/09/03
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             DTYPE,       ICNTE,       ICNTS
      INTEGER             IETRN(NDTRN),             ISTRN(NDTRN)
      INTEGER             ITRN,        MAXT,        NDTEM,       NDTIN
      INTEGER             NDTRN,       NPSPL,       NV
      LOGICAL             LTRNG(NOUT)
      REAL*8              GAMMA(NDTEM),             GAMOSA(NPSPL)
      REAL*8              SCOM(NDTEM,NDTRN),        TOA(NDTIN)
      REAL*8              TOSA(NPSPL), USCEF(NDTEM)
```

8.22 h9tran: Subroutine h9tran from library adas8xx

```
       SUBROUTINE H9TRAN( NDLEV  , NDTRN  , NDTEM ,
     &                    IL     , ISTRN  , NV    ,
     &                    IA     , WA     , XJA   ,
     &                    I1A    , I2A    , AVAL  ,
     &                    SCOM,
     &                    IUPPER , ILOWER ,
     &                    LUPPER , LLOWER ,
     &                    WUPPER , WLOWER ,
     &                    EUPPER , ELOWER ,
     &                    AA     , GAMMA
     &                  )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: H9TRAN *********************
C
C  PURPOSE:  TO SET UP SELECTED TRANSITION PARAMETERS.
C
C  CALLING PROGRAM: ADAS809
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  NDLEV     = MAXIMUM NUMBER OF INDEX LEVELS
C  INPUT : (I*4)  NDTRN     = MAXIMUM NUMBER OF TRANSITIONS
C  INPUT : (I*4)  NDTEM     = MAXIMUM NUMBER OF INPUT FILE TEMPERATURES
C
C  INPUT : (I*4)  IL        = NUMBER OF INDEX LEVELS
C  INPUT : (I*4)  ISTRN     = SELECTED TRANSITION INDEX.
C  INPUT : (I*4)  NV        = INPUT DATA FILE: NUMBER OF GAMMA/TEMPERATURE
C                             PAIRS FOR THE SELECTED TRANSITION.
C
C  INPUT : (I*4)  IA()      = LEVEL INDEX NUMBER ARRAY
C  INPUT : (R*8)  WA()      = LEVEL ENERGIES RELATIVE TO LEVEL 1 (CM-1)
C  INPUT : (R*8)  XJA()     = QUANTUM NUMBER (J-VALUE) FOR LEVEL
C                             NOTE: (2*XJA)+1 = STATISTICAL WEIGHT
C
C  INPUT : (I*4)  I1A()     = LOWER LEVEL INDEX FOR ELECTRON IMPACT
C                             TRANSITION
C  INPUT : (I*4)  I2A()     = UPPER LEVEL INDEX FOR ELECTRON IMPACT
C                             TRANSITION
C  INPUT : (I*4)  AVAL()    = A-VALUE FOR ELECTRON IMPACT TRANSITION
C  INPUT : (I*4)  SCOM(,)   = GAMMA VALUES FOR ELECTRON IMPACT
C                             (DE-)EXCITATION
C                             1st DIMENSION: TEMPERATURE INDEX
C                             2nd DIMENSION: TRANSITION INDEX
C  OUTPUT: (I*4)  IUPPER    = SELECTED TRANSITION: UPPER LEVEL ARRAY INDEX
C  OUTPUT: (I*4)  ILOWER    = SELECTED TRANSITION: LOWER LEVEL ARRAY INDEX
C
C
C  OUTPUT: (I*4)  LUPPER    = SELECTED TRANSITION: UPPER INDEX LEVEL
C  OUTPUT: (I*4)  LLOWER    = SELECTED TRANSITION: LOWER INDEX LEVEL
C
C  OUTPUT: (R*8)  WUPPER    = SELECTED TRANSITION: UPPER LEVEL STAT. WT.
C  OUTPUT: (R*8)  WLOWER    = SELECTED TRANSITION: LOWER LEVEL STAR. WT.
C                             (NOTE: STAT. WT. = STATISTICAL WEIGHT)
C
C  OUTPUT: (R*8)  EUPPER    = SELECTED TRANSITION: UPPER ENERGY LEVEL
C                             RELATIVE TO INDEX LEVEL 1. (CM-1)
C  OUTPUT: (R*8)  ELOWER    = SELECTED TRANSITION: LOWER ENERGY LEVEL
C                             RELATIVE TO INDEX LEVEL 1. (CM-1)
```

```
C  OUTPUT: (R*8)  AA        = SELECTED TRANSITION A-VALUE (SEC-1)
C  OUTPUT: (R*8)  GAMMAUP() = INPUT DATA FILE: SELECTED EXCITATION -
C                                               GAMMAUP VALUE AT 'TEMP()'
C  OUTPUT: (R*8)  GAMMADN() = INPUT DATA FILE: SELECTED DE-EXCITATION -
C                                               GAMMADN VALUE AT 'TEMP()'
C
C          (I*4)  I         = GENERAL USE.
C
C ROUTINES: NONE
C
C AUTHOR:  HUGH SUMMERS (UNIVERSITY OF STRATHCLYDE)
C          JA7.08
C          EXT. 4196
C
C DATE:    30/11/01
C
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      INTEGER             I1A(NDTRN),  I2A(NDTRN),  IA(NDLEV),   IL
      INTEGER             ILOWER,      ISTRN,       IUPPER,      LLOWER
      INTEGER             LUPPER,      NDLEV,       NDTEM,       NDTRN
      INTEGER             NV
      REAL*8              AA,          AVAL(NDTRN), ELOWER,      EUPPER
      REAL*8              GAMMA(NDTEM),             SCOM(NDTEM,NDTRN)
      REAL*8              WA(NDLEV),   WLOWER,      WUPPER
      REAL*8              XJA(NDLEV)
```

8.23 h9trni: Subroutine h9trni from library adas8xx

```
       SUBROUTINE H9TRNI( NDLEV  , NDTRN  , NDTEM , ndmet ,
     &                    IL     , ISTRN  , NV    ,
     &                    IA     , WA     , XJA   ,
     &                    I1A    , I2A    , AVAL  ,
     &                    SCOM   , zpla   , bwnoa , ipla ,
     &                    IUPPER , ILOWER ,
     &                    LUPPER , LLOWER ,
     &                    WUPPER , WLOWER ,
     &                    EUPPER , ELOWER ,
     &                    AA     , GAMMA  ,
     &                    zeta   , ip
     &                  )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: H9TRNI *********************
C
C  PURPOSE:  TO SET UP SELECTED IONISATION TRANSITION PARAMETERS
C
C  CALLING PROGRAM: ADAS809
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  NDLEV     = MAXIMUM NUMBER OF INDEX LEVELS
C  INPUT : (I*4)  NDTRN     = MAXIMUM NUMBER OF TRANSITIONS
C  INPUT : (I*4)  NDTEM     = MAXIMUM NUMBER OF INPUT FILE TEMPERATURES
C
C  INPUT : (I*4)  IL        = NUMBER OF INDEX LEVELS
C  INPUT : (I*4)  ISTRN     = SELECTED TRANSITION INDEX.
C  INPUT : (I*4)  NV        = INPUT DATA FILE: NUMBER OF GAMMA/TEMPERATURE
C                             PAIRS FOR THE SELECTED TRANSITION.
C
C  INPUT : (I*4)  IA()      = LEVEL INDEX NUMBER ARRAY
C  INPUT : (R*8)  WA()      = LEVEL ENERGIES RELATIVE TO LEVEL 1 (CM-1)
C  INPUT : (R*8)  XJA()     = QUANTUM NUMBER (J-VALUE) FOR LEVEL
C                             NOTE: (2*XJA)+1 = STATISTICAL WEIGHT
C
C  INPUT : (I*4)  I1A()     = LOWER LEVEL INDEX FOR ELECTRON IMPACT
C                             TRANSITION
C  INPUT : (I*4)  I2A()     = UPPER LEVEL INDEX FOR ELECTRON IMPACT
C                             TRANSITION
C  INPUT : (I*4)  AVAL()    = A-VALUE FOR ELECTRON IMPACT TRANSITION
C  INPUT : (I*4)  SCOM(,)   = GAMMA VALUES FOR ELECTRON IMPACT
C                             (DE-)EXCITATION
C                             1st DIMENSION: TEMPERATURE INDEX
C                             2nd DIMENSION: TRANSITION INDEX
C  OUTPUT: (I*4)  IUPPER    = SELECTED TRANSITION: UPPER LEVEL ARRAY INDEX
C  OUTPUT: (I*4)  ILOWER    = SELECTED TRANSITION: LOWER LEVEL ARRAY INDEX
C
C
C  OUTPUT: (I*4)  LUPPER    = SELECTED TRANSITION: UPPER INDEX LEVEL
C  OUTPUT: (I*4)  LLOWER    = SELECTED TRANSITION: LOWER INDEX LEVEL
C
C  OUTPUT: (R*8)  WUPPER    = SELECTED TRANSITION: UPPER LEVEL STAT. WT.
C  OUTPUT: (R*8)  WLOWER    = SELECTED TRANSITION: LOWER LEVEL STAR. WT.
C                             (NOTE: STAT. WT. = STATISTICAL WEIGHT)
C
C  OUTPUT: (R*8)  EUPPER    = SELECTED TRANSITION: UPPER ENERGY LEVEL
C                             RELATIVE TO INDEX LEVEL 1. (CM-1)
C  OUTPUT: (R*8)  ELOWER    = SELECTED TRANSITION: LOWER ENERGY LEVEL
```

```
C                               RELATIVE TO INDEX LEVEL 1. (CM-1)
C  OUTPUT: (R*8)  AA        = SELECTED TRANSITION A-VALUE (SEC-1)
C  OUTPUT: (R*8)  GAMMAUP() = INPUT DATA FILE: SELECTED EXCITATION -
C                                               GAMMAUP VALUE AT 'TEMP()'
C  OUTPUT: (R*8)  GAMMADN() = INPUT DATA FILE: SELECTED DE-EXCITATION -
C                                               GAMMADN VALUE AT 'TEMP()'
C
C          (I*4)  I         = GENERAL USE.
C
C ROUTINES: NONE
C
C AUTHOR:  HUGH SUMMERS (UNIVERSITY OF STRATHCLYDE)
C          JA7.08
C          EXT. 4196
C
C DATE:    30/11/01
C
C UPDATE:  Paul Bryans
C    24/11/04
C    Added extra parameters needed for ionisation transition
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      INTEGER             I1A(NDTRN),  I2A(NDTRN),  IA(NDLEV),   IL
      INTEGER             ILOWER,      IPLA(NDMET,NDLEV),        ISTRN
      INTEGER             IUPPER,      LLOWER,      LUPPER,      NDLEV
      INTEGER             NDMET,       NDTEM,       NDTRN,       NV
      REAL*8              AA,          AVAL(NDTRN), BWNOA(NDMET)
      REAL*8              ELOWER,      EUPPER,      GAMMA(NDTEM)
      REAL*8              IP,          SCOM(NDTEM,NDTRN)
      REAL*8              WA(NDLEV),   WLOWER,      WUPPER
      REAL*8              XJA(NDLEV),  ZETA,        ZPLA(NDMET,NDLEV)
```

8.24 h9ttyp: Subroutine h9ttyp from library adas8xx

```
      SUBROUTINE H9TTYP( NDLEV  , NDMET  , NDTRN  , NPLR  , NPLI  ,
     &                   ITRAN  , TCODE  , I1A    , I2A   , AVAL  ,
     &  IADFTYP,
     &                   ICNTE  , ICNTP  , ICNTR  , ICNTH , ICNTI ,
     &                   ICNTL  , ICNTS  ,
     &                   IETRN  , IPTRN  , IRTRN  , IHTRN , IITRN ,
     &                   ILTRN  , ISTRN  ,
     &                                     IE1A   , IE2A  , AA    ,
     &                                     IP1A   , IP2A  ,
     &                                     IA1A   , IA2A  , AUGA  ,
     &                                     IL1A   , IL2A  , WVLA  ,
     &                                     IS1A   , IS2A  , LSS04A
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: BXTTYP *********************
C
C  PURPOSE:  TO SORT TRANSITION ARRAYS INTO SEVEN TRANSITION/RECOMB TYPES
C
C  CALLING PROGRAM: General
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  NDLEV   = MAXIMUM NUMBER OF LEVELS THAT CAN BE READ
C  INPUT : (I*4)  NDMET   = MAXIMUM NUMBER OF METASTABLES
C  INPUT : (I*4)  NDTRN   = MAXIMUM NUMBER OF TRANS. THAT CAN BE READ
C
C  OUTPUT: (I*4)  NPLR    = NO. OF ACTIVE METASTABLES OF (Z+1) ION
C  OUTPUT: (I*4)  NPLI    = NO. OF ACTIVE METASTABLES OF (Z-1) ION
C
C  INPUT : (I*4)  ITRAN   = INPUT DATA FILE: NUMBER OF TRANSITIONS
C  INPUT : (C*1)  TCODE() = TRANSITION: DATA TYPE POINTER:
C                           ' ','1','2','3' => Electron Impact Transition
C                           'P','p' => Proton   Impact   Transition
C                           'H','h' => Charge   Exchange Recombination
C                           'R','r' => Free     Electron Recombination
C                           'I','i' => Electron Impact Ionisation to z
C                           'L','l' => Satellites from DR Recombination
C                           'S','s' => Electron Impact Ionisation to z+1
C  INPUT : (I*4)  I1A()   = TRANSITION:
C                            LOWER ENERGY LEVEL INDEX (CASE ' ' & 'P')
C                            PARENT ENERGY LEVEL INDEX(CASE 'H' & 'R')
C                                                     (         & 'L')
C                            FINAL PARENT LEVEL INDEX (CASE 'S')
C  INPUT : (I*4)  I2A()   = TRANSITION:
C                            UPPER ENERGY LEVEL INDEX (CASE ' ' & 'P')
C                            CAPTURING    LEVEL INDEX (CASE 'H' & 'R')
C                                                     (         & 'L')
C                            IONISING     LEVEL INDEX (CASE 'S')
C  INPUT : (R*8)  AVAL()  = TRANSITION:
C                            A-VALUE (SEC-1)          (CASE ' ')
C                            NEUTRAL BEAM ENERGY      (CASE 'H')
C                            AUGER VALUE(SEC-1)       (CASE 'R')
C                            PARENT WAVLENGTH (A)     (CASE 'L')
C                            NOT USED                 (CASE 'P' & 'S')
C
C  OUTPUT: (I*4)  ICNTE   = NUMBER OF ELECTRON IMPACT TRANSITIONS INPUT
C  OUTPUT: (I*4)  ICNTP   = NUMBER OF PROTON IMPACT TRANSITIONS INPUT
C  OUTPUT: (I*4)  ICNTR   = NUMBER OF FREE ELECTRON RECOMBINATIONS INPUT
```

```
C  OUTPUT: (I*4)  ICNTH   = NO. OF CHARGE EXCHANGE RECOMBINATIONS INPUT
C  OUTPUT: (I*4)  ICNTI   = NO. OF IONISATIONS TO Z INPUT
C  OUTPUT: (I*4)  ICNTL   = NO. OF SATELLITE DR RECOMBINATIONS INPUT
C  OUTPUT: (I*4)  ICNTS   = NO. OF IONISATIONS TO Z+1 INPUT
C
C  OUTPUT: (I*4)  IETRN() = ELECTRON IMPACT TRANSITION:
C                           INDEX VALUES IN MAIN TRANSITION ARRAYS WHICH
C                           REPRESENT ELECTRON IMPACT TRANSITIONS.
C  OUTPUT: (I*4)  IPTRN() = PROTON IMPACT TRANSITION:
C                           INDEX VALUES IN MAIN TRANSITION ARRAYS WHICH
C                           REPRESENT PROTON IMPACT TRANSITIONS.
C  OUTPUT: (I*4)  IRTRN() = FREE ELECTRON RECOMBINATION:
C                           INDEX VALUES IN MAIN TRANSITION ARRAYS WHICH
C                           REPRESENT FREE ELECTRON RECOMBINATIONS.
C  OUTPUT: (I*4)  IHTRN() = CHARGE EXCHANGE RECOMBINATION:
C                           INDEX VALUES IN MAIN TRANSITION ARRAYS WHICH
C                           REPRESENT CHARGE EXCHANGE RECOMBINATIONS.
C  OUTPUT: (I*4)  IITRN() = ELECTRON IMPACT IONISATION:
C                           INDEX VALUES IN MAIN TRANSITION ARRAYS WHICH
C                           REPRESENT IONISATIONS FROM LOWER STAGE ION.
C  OUTPUT: (I*4)  ILTRN() = SATELLITE DR RECOMBINATION:
C                           INDEX VALUES IN MAIN TRANSITION ARRAYS WHICH
C                           REPRESENT SATELLITE DR RECOMBINATIONS.
C  OUTPUT: (I*4)  ISTRN() = ELECTRON IMPACT IONISATION:
C                           INDEX VALUES IN MAIN TRANSITION ARRAYS WHICH
C                           REPRESENT IONISATIONS TO UPPER STAGE ION.
C
C  OUTPUT: (I*4)  IE1A()  = ELECTRON IMPACT TRANSITION:
C                            LOWER ENERGY LEVEL INDEX
C  OUTPUT: (I*4)  IE2A()  = ELECTRON IMPACT TRANSITION:
C                            UPPER ENERGY LEVEL INDEX
C  OUTPUT: (R*8)  AA()    = ELECTRON IMPACT TRANSITION: A-VALUE (SEC-1)
C
C
C  OUTPUT: (I*4)  IP1A()  = PROTON IMPACT TRANSITION:
C                            LOWER ENERGY LEVEL INDEX
C  OUTPUT: (I*4)  IP2A()  = PROTON IMPACT TRANSITION:
C                            UPPER ENERGY LEVEL INDEX
C
C  OUTPUT: (I*4)  IA1A()  = AUGER TRANSITION:
C                            PARENT ENERGY LEVEL INDEX
C  OUTPUT: (I*4)  IA2A()  = AUGER TRANSITION:
C                            RECOMBINED ION ENERGY LEVEL INDEX
C  OUTPUT: (R*8)  AUGA()  = AUGER TRANSITION: AUG-VALUE (SEC-1)
C                            RECOMBINED ION ENERGY LEVEL INDEX
C  OUTPUT: (I*4)  IL1A()  = SATELLITE DR TRANSITION:
C                            RECOMBINING ION  INDEX
C  OUTPUT: (I*4)  IL2A()  = SATELLITE DR TRANSITION:
C                            RECOMBINED ION INDEX
C  OUTPUT: (R*8)  WVLA()  = SATELLITE DR TRANSITION: PARENT WVLGTH.(A)
C                            DR SATELLITE LINE INDEX
C  OUTPUT: (I*4)  IS1A()  = IONISING TRANSITION:
C                            IONISED ION  INDEX
C  OUTPUT: (I*4)  IS2A()  = IONISING TRANSITION:
C                            IONISING ION INDEX
C  OUTPUT: (L*4)  LSS04A(,)= .TRUE. => IONIS. RATE SET IN ADF04 FILE:
C                            .FALSE.=> NOT SET IN ADF04 FILE
C                            1ST DIM: LEVEL INDEX
C                            2ND DIM: PARENT METASTABLE INDEX
C
C          (I*4)  I       = GENERAL USE.
```

```
C
C
C ROUTINES: NONE
C
C AUTHOR:  HP SUMMERS (REVISION OF BXTTYP BY PE BRIDEN)
C          K1/1/57
C          JET EXT. 4941
C
C DATE  :  11/06/92
C
C-----------------------------------------------------------------------
C PUT UNDER SCCS CONTROL:
C
C VERSION: 1.1 DATE: 10/05/96
C MODIFIED: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C     - FIRST PUT UNDER SCCS
C
C VERSION: 1.2 DATE: 13/09/99
C MODIFIED: HUGH SUMMERS, UNIVERSITY OF STRATHCLYDE
C     - ADDED DETECTION OF L-LINES AND S-LINES
C
C-----------------------------------------------------------------------
C
C VERSION: 1.2 DATE: 01/05/2003
C MODIFIED: Martin O'Mullane
C     - Replaced original bxttyp with b8ttyp version 1.2.
C Hence the 1.2 version no.
C
C-----------------------------------------------------------------------
C
C VERSION: 1.3 DATE: 27/11/03
C MODIFIED: Paul Bryans, University of Strathclyde
C           - allow lower case values of tcode and different types of
C               electron impact excitation (i.e. ' ','1','2','3')
C     - fix number of electron impact transitions for adf04 iv
C
C-----------------------------------------------------------------------
C VERSION: 1.4 DATE: 26/11/04
C MODIFIED: Allan Whiteford, University of Strathclyde
C           - Renamed to h9ttyp.
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      CHARACTER           TCODE(NDTRN)
      INTEGER             I1A(NDTRN),  I2A(NDTRN),  IA1A(NDTRN)
      INTEGER             IA2A(NDTRN), IADFTYP,     ICNTE,       ICNTH
      INTEGER             ICNTI,       ICNTL,       ICNTP,       ICNTR
      INTEGER             ICNTS,       IE1A(NDTRN), IE2A(NDTRN)
      INTEGER             IETRN(NDTRN),             IHTRN(NDTRN)
      INTEGER             IITRN(NDTRN),             IL1A(NDLEV)
      INTEGER             IL2A(NDLEV), ILTRN(NDTRN)
      INTEGER             IP1A(NDTRN), IP2A(NDTRN), IPTRN(NDTRN)
      INTEGER             IRTRN(NDTRN),             IS1A(NDLEV)
      INTEGER             IS2A(NDLEV), ISTRN(NDTRN),             ITRAN
      INTEGER             NDLEV,       NDMET,       NDTRN,       NPLI
      INTEGER             NPLR
      LOGICAL             LSS04A(NDLEV,NDMET)
      REAL*8              AA(NDTRN),   AUGA(NDTRN), AVAL(NDTRN)
      REAL*8              WVLA(NDLEV)
```

8.25 h9wr11: Subroutine h9wr11 from library adas8xx

```
       SUBROUTINE H9WR11( IUNT11 , DSNINP ,
     &                    NDLEV  , NDTRN , NVMAX , NDMET ,
     &                    DATE   ,
     &                    TITLED , IZ     , IZ0   , IZ1   , BWNOA ,
     &                    IL     ,
     &                    CSTRGA , ISA   , ILA   , XJA   , WA ,
     &                    IPLA   , ZPLA  ,
     &                    NVN    , SCEFN  ,
     &                    icnte  , icnts  , ietrn , istrn ,
     &                    TCODE  , I1A    , I2A   , AVAL  , SCOMUP ,
     &                    SCOMDN , DTYPE  , KAPPA , ADF37 , druval ,
     &                    lbeth  , beth   , npl   , cprta
     &                  )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: H9WR11 *********************
C
C  PURPOSE:  PRODUCES AN ADF04 TYPE 3 FILE, WHERE THE CONTENTS IS
C            CONSIDERED AS THE OUTPUT DATA SET FROM ADAS809.
C
C  CALLING PROGRAM: ADAS809
C
C
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  IUNT11  = UNIT TO WHICH INPUT FILE IS ALLOCATED
C  INPUT : (C*80) DSNINP  = NAME OF INPUT ADF04 FILE
C  INPUT : (I*4)  NDLEV   = MAXIMUM NUMBER OF LEVELS
C  INPUT : (I*4)  NDTRN   = MAX. NUMBER OF TRANSITIONS
C  INPUT : (I*4)  NVMAX   = MAX. NUMBER OF TEMPERATURES
C  INPUT : (I*4)  NDMET   = MAX. NUMBER OF METASTABLES
C  INPUT : (C*8)  DATE    = DATE (AS DD/MM/YY).
C  INPUT : (C*3)  TITLED  = ELEMENT SYMBOL.
C  INPUT : (I*4)  IZ      =  RECOMBINED ION CHARGE READ
C  INPUT : (I*4)  IZ0     =         NUCLEAR CHARGE READ
C  INPUT : (I*4)  IZ1     = RECOMBINING ION CHARGE READ
C                           (NOTE: IZ1 SHOULD EQUAL IZ+1)
C  INPUT : (R*8)  BWNO    = IONISATION POTENTIAL (CM-1)
C
C  INPUT : (I*4)  IL      = INPUT DATA FILE: NUMBER OF ENERGY LEVELS
C
C  INPUT : (C*18) CSTRGA()= NOMENCLATURE/CONFIGURATION FOR LEVEL 'IA()'
C  INPUT : (I*4)  ISA()   = MULTIPLICITY FOR LEVEL 'IA()'
C                           NOTE: (ISA-1)/2 = QUANTUM NUMBER (S)
C  INPUT : (I*4)  ILA()   = QUANTUM NUMBER (L) FOR LEVEL 'IA()'
C  INPUT : (R*8)  XJA()   = QUANTUM NUMBER (J-VALUE) FOR LEVEL 'IA()'
C                           NOTE: (2*XJA)+1 = STATISTICAL WEIGHT
C  INPUT : (R*8)  WA()    = ENERGY RELATIVE TO LEVEL 1 (CM-1) FOR LEVEL
C                           'IA()'
C  INPUT : (I*4)  IPLA(,) = PARENT INDEX FOR CONTRIBUTION TO IONIS.
C                           OF LEVEL
C                           1ST DIMENSION: PARENT INDEX
C                           2ND DIMENSION: LEVEL INDEX
C  INPUT : (R*8)  ZPLA(,) = EFF. ZETA PARAM. FOR CONTRIBUTION TO IONIS.
C                           OF LEVEL
C                           1ST DIMENSION: PARENT INDEX
C                           2ND DIMENSION: LEVEL INDEX
C
```

```
C  INPUT : (I*4)  NVN     = INPUT DATA FILE: NUMBER OF GAMMA/TEMPERATURE
C                           PAIRS FOR A GIVEN TRANSITION.
C  INPUT : (R*8)  SCEFN() = INPUT DATA FILE: ELECTRON TEMPERATURES (K)
C                           (INITIALLY JUST THE MANTISSA. SEE 'ITPOW()')
C                           (NOTE: TE=TP=TH IS ASSUMED)
C
C  INPUT : (I*4)  ITRNE   = INPUT DATA FILE: NO OF EXCITATION TRANSITIONS
C  INPUT : (I*4)  ITRNS   = INPUT DATA FILE: NO OF IONISATION TRANSITIONS
C
C  INPUT : (C*1)  TCODE() = TRANSITION: DATA TYPE POINTER:
C                           ' ' => Electron Impact   Transition
C                           'P' => Proton   Impact   Transition
C                           'H' => Charge   Exchange Recombination
C                           'R' => Free     Electron Recombination
C  INPUT : (I*4)  I1A()   = TRANSITION:
C                            LOWER ENERGY LEVEL INDEX (CASE ' ' & 'P')
C                            SIGNED PARENT      INDEX (CASE 'H' & 'R')
C  INPUT : (I*4)  I2A()   = TRANSITION:
C                            UPPER ENERGY LEVEL INDEX (CASE ' ' & 'P')
C                            CAPTURING    LEVEL INDEX (CASE 'H' & 'R')
C  INPUT : (R*8)  AVAL()  = TRANSITION:
C                            A-VALUE (SEC-1)          (CASE ' ')
C                            NEUTRAL BEAM ENERGY      (CASE 'H')
C                            NOT USED                 (CASE 'P' & 'R')
C  INPUT : (R*8)  SCOMN(,)= TRANSITION:
C                            GAMMA VALUES             (CASE ' ' & 'P')
C                            RATE COEFFT. (CM3 SEC-1) (CASE 'H' & 'R')
C                           1ST DIMENSION - TEMPERATURE 'SCEF()'
C                           2ND DIMENSION - TRANSITION NUMBER
C
C
C  ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          XXNAME     ADAS      FINDS REAL NAME OF USER
C          XXWSTR     ADAS      WRITES STRING TO A UNIT WITH TRAILING
C                               BLANKS REMOVED
C
C
C  AUTHOR : H. P. SUMMERS, UNIVERSITY OF STRATHCLYDE
C           JA8.08
C           TEL.  0141-553-4196
C
C  DATE:    30/11/01
C
C  UPDATE:
CC
C-----------------------------------------------------------------------
      CHARACTER*80        ADF37
      CHARACTER*9         CPRTA(NDMET)
      CHARACTER*18        CSTRGA(NDLEV)
      CHARACTER*8         DATE
      CHARACTER*80        DSNINP
      CHARACTER           TCODE(NDTRN)
      CHARACTER*3         TITLED
      INTEGER             DTYPE,       I1A(NDTRN),  I2A(NDTRN),  ICNTE
      INTEGER             ICNTS,       IETRN(NDTRN),             IL
      INTEGER             ILA(NDLEV),  IPLA(NDMET,NDLEV)
      INTEGER             ISA(NDLEV),  ISTRN(NDTRN),             IUNT11
      INTEGER             IZ,          IZ0,         IZ1,         NDLEV
      INTEGER             NDMET,       NDTRN,       NPL,         NVMAX
```

```
INTEGER            NVN
LOGICAL            LBETH
REAL*8             AVAL(NDTRN), BETH(NDTRN), BWNOA(NDMET)
REAL*8             DRUVAL,      KAPPA,       SCEFN(NVMAX)
REAL*8             SCOMDN(NVMAX,NDTRN),      SCOMUP(NVMAX,NDTRN)
REAL*8             WA(NDLEV),   XJA(NDLEV),  ZPLA(NDMET,NDLEV)
```

8.26 haadas2: Subroutine haadas2 from library adas8xx

```
C
       subroutine haadas2(ichan   , date   ,
     +                    ntdim   , nnedim ,
     +                    data    ,
     +                    itmax   , idmax  ,
     +                    dense   , tempe  ,
     +                    iz0     , iz1    ,
     +                    name    , method ,
     +                    cstrg1  , cstrg2 ,
     +                    infplt  , comments , ncomments )
c
c
c-----------------------------------------------------------------------
c
c  ******************** fortran77 subroutine: haadas2 *****************
c
c  purpose : to write plt standard adas density dependent data
c            The data is in the form :-
c                        data(it,id,iz)
c            where,
c                  it     :  temperature index ( 1 - itmax )
c                  id     :  density     index ( 1 - idmax )
c
c            with electron temperatures ---- tempe(1 - itmax)
c                electron densities     ---- dense(1 - idmax)
c
c  calling program: adas810
c
c  input :  (i*4)  ichan     =  stream number (previously allocated)
c  input :  (i*4)  ntdim     =  maximum number of temperatures
c  input :  (i*4)  nnedim    =  maximum number of densities
c  input :  (i*4)  itmax     =  number of temperatures
c  input :  (i*4)  idmax     =  number of densities
c  input :  (i*4)  iz0       =  nuclear charge of species
c  input :  (i*4)  iz1       =  ion charge +1
c  input :  (i*4)  ncomments =  number of comment strings
c  input :  (r*8)  data       =  profile array (see above)
c  input :  (r*8)  dense      =  electron densities
c  input :  (r*8)  tempe      =  electron temperatures
c  input :  (c*13) name       =  name of element
c  input :  (c*25) method     =  method used in the calculations
c  input :  (c*10) cstrg1     =  ground information string
c  input :  (c*10) cstrg2     =  parent information string
c  input :  (c*8)  infplt     =  total line power stringssection
c  input :  (c*80) comments() =  comment strings
c
c
c author:  h. p. summers, university of strathlcyde
c
c date:    24 April 2002
c
c update:
c
c-----------------------------------------------------------------------
      CHARACTER*80        COMMENTS(NCOMMENTS)
      CHARACTER*10        CSTRG1,      CSTRG2
      CHARACTER*8         DATE,        INFPLT
      CHARACTER*24        METHOD
      CHARACTER*13        NAME
```

```
INTEGER             ICHAN,       IDMAX,       ITMAX,       IZ0
INTEGER             IZ1,         NCOMMENTS,   NNEDIM,      NTDIM
REAL*8              DATA(NTDIM,NNEDIM),       DENSE(NNEDIM)
REAL*8              TEMPE(NTDIM)
```

8.27 hapecf: Subroutine hapecf from library adas8xx

```
      subroutine hapecf( iunt15 , iunt40 , iunt11 , iunt11f , iunt35  ,
     &                   open15 , open40 , open11 , open11f , open35  ,
     &                   dsn35  ,
     &                   ndlev  , ndtrn  , ndtem  , ndden   , ndmet   ,
     &                   ndwvl  , ndpix  ,
     &                   nmet   , imetr  , nord   , iordr   ,
     &                   maxt   , teva   ,
     &                   maxd   , densa  ,
     &                   lpsel  , lzsel  , liosel ,
     &                   lhsel  , lrsel  , lisel  , lnsel   ,
     &                   iz     , iz0    , iz1    ,
     &                   npl    , bwno   ,
     &                   nplr   , npli   , npl3   ,
     &                   dsninc , dsnexp ,
     &                   titled , date   , user   ,
     &                   il     ,
     &                   ia     , cstrga , isa    , ila     , xja , wa ,
     &                   icnte  , icntr  , icnth  , icnti   ,
     &                   ietrn  ,
     &                   ie1a   , ie2a   , aa     ,
     &                   lnorm  ,
     &                   stckm  , stvr   , stvi   , stvh    ,
     &                   stvrm  , stvim  , stvhm  ,
     &                   ratpia , ratmia , stack  ,
     &                   lsseta , lss04a ,
     &                   nwvl   , npix   , wvmin  , wvmax   , avlt
     &                 )
c-----------------------------------------------------------------------
c
c  ******************** fortran77 subroutine: hapecf ******************
c
c  purpose:  to prepare pec, envelope feature f-pec, plt and plt-filter
c            passing files for diagnostic use.
c
c  calling program: adas810
c
c
c  subroutine:
c
c  input : (i*4)  iunt15  = output unit for adf15 pec results
c  input : (i*4)  iunt40  = output unit for adf40 fpec results
c  input : (i*4)  iunt11  = output unit for adf11 plt results
c  input : (i*4)  iunt11f = output unit for adf41 filtered plt results
c  input : (i*4)  iunt35  = input  unit for adf35 filter file
c
c  input : (i*4)  ndlev   = maximum number of levels allowed
c  input : (i*4)  ndtrn   = maximum number of transitions allowed
c  input : (i*4)  ndtem   = maximum number of temperatures allowed
c  input : (i*4)  ndden   = maximum number of densities allowed
c  input : (i*4)  ndmet   = maximum number of metastables allowed
c  input : (i*4)  ndwvl   = maximum number of wavelength intervals
c  input : (i*4)  ndpix   = maximum no. of pixels in a wvl. interval
c
c  input : (i*4)  nmet    = number of metastables levels: 1<=nmet<=ndmet
c  input : (i*4)  imetr() = index of metastable in complete level list
c  input : (i*4)  nord    = number of ordinary levels ('il' - 'nmet')
c  input : (i*4)  iordr() = index of ordinary levels in complete level
c                           list.
c
```

```
c  input : (i*4)  maxt    = number of input temperatures ( 1 -> 'ndtem')
c  input : (i*4)  maxd    = number of input densities ( 1 -> 'ndden')
c  input : (r*8)  teva()  = electron temperatures (units: ev)
c  input : (r*8)  densa() = electron densities  (units: cm-3)
c
c  input : (l*4)  lpsel    = .true.  => include proton collisions
c                          = .false. =>do not include proton collisions
c  input : (l*4)  lzsel    = .true.  => scale proton collisions with
c                                       plasma z effective'zeff'.
c                          = .false. => do not scale proton collisions
c                                       with plasma z effective 'zeff'.
c                           (only used if 'lpsel=.true.')
c  input   (l*4)  liosel  = .true.  => include ionisation rates
c                         = .false. => do not include ionisation rates
c                            for recom and  3-body
c  input   (l*4)  lhsel   = .true.  => include charge transfer from
c                                      neutral hydrogren.
c                         = .false. => do not include charge transfer
c                                      from neutral hydrogren.
c  input   (l*4)  lrsel   = .true.  => include free electron
c                                      recombination.
c                         = .false. => do not include free electron
c                                      recombination.
c  input   (l*4)  lisel   = .true.  => include electron impact
c                                      ionisation.
c                         = .false. => do not include free electron
c                                      recombination.
c  input : (l*4)  lnsel    = .true.  => include projected bundle-n data
c                                       from datafile if available
c                          = .false. => do not include projected
c                                       bundle-n data
c
c  input : (i*4)  iz      =  recombined ion charge read
c  input : (i*4)  iz0     =         nuclear charge read
c  input : (i*4)  iz1     = recombining ion charge read
c                           (note: iz1 should equal iz+1)
c
c  input : (i*4)  npl      = no. of metastables of(z+1) ion accessed
c                            by excited state ionisation in copase
c                            file with ionisation potentials given
c                            on the first data line
c  input : (r*8)  bwno    = ionisation potential (cm-1)
c
c  input : (i*4)  nplr     = no. of active metastables of (z+1) ion
c  input : (i*4)  npli     = no. of active metastables of (z-1) ion
c  input : (i*4)  npl3     = no. of active metastables of (z+1) ion with
c                            three-body recombination on.
c
c  input : (c*44) dsninc  = input copase data set name
c  input : (c*80) dsnexp  = input expansion file
c
c  input : (c*3)  titled  = element symbol.
c  input : (c*8)  date    = current date.
c  input : (c*30) user    = full name of author.
c
c  input : (i*4)  il      = number of energy levels
c
c  input : (i*4)  ia()    = energy level index number
c  input : (c*18) cstrga()= nomenclature/configuration for level 'ia()'
c  input : (i*4)  isa()   = multiplicity for level 'ia()'
c                           note: (isa-1)/2 = quantum number (s)
```

```
c  input : (i*4)  ila()    = quantum number (l) for level 'ia()'
c  input : (r*8)  xja()    = quantum number (j-value) for level 'ia()'
c                            note: (2*xja)+1 = statistical weight
c  input : (r*8)  wa()     = energy relative to level 1 (cm-1)
c                            dimension: level index
c
c
c  input : (i*4)  icnte    = number of electron impact transitions input
c  input : (i*4)  icntr    = number of free electron recombinations input
c  input : (i*4)  icnth    = no. of charge exchange recombinations input
c  input : (i*4)  icnti    = number of lower stage ionisations      input
c
c  input : (i*4)  ietrn() = electron impact transition:
c                            index values in main transition arrays which
c  input : (i*4)  ie1a()   = electron impact transition:
c                             lower energy level index
c  input : (i*4)  ie2a()   = electron impact transition:
c                             upper energy level index
c  input : (r*8)  aa()     = electron impact transition: a-value (sec-1)
c
c  input : (l*4)  lnorm    =.true.  => if nmet=1 then various
c                                      emissivity output files
c                                      normalised to stage tot.populatn.
c                                      (** norm type = t)
c                          =.false. => otherwise normalise to identified
c                                      metastable populations.
c                                       (** norm type = m)
c
c
c  input : (r*8) stckm(,,) = metastable populations stack
c                             1st dimension: metastable index
c                             2nd dimension: temperature index
c                             3rd dimension: density index
c  input : (r*4) stvr(,,,) = free electron recombination coefficients
c                             1st dimension: ordinary level index
c                             2nd dimension: temperature index
c                             3rd dimension: density index
c                             4th dimension: parent index
c  input : (r*4) stvi(,,,) = electron impact ionisation  coefficients
c                             1st dimension: ordinary level index
c                             2nd dimension: temperature index
c                             3rd dimension: density index
c                             4th dimension: parent index
c  input : (r*4) stvh(,,,) =  charge exchange coefficients
c                             1st dimension: ordinary level index
c                             2nd dimension: temperature index
c                             3rd dimension: density index
c  input : (r*8) stvrm(,,,)= metastable free electron recombination
c                            coefficients.
c                             1st dimension: metastable index
c                             2nd dimension: temperature index
c                             3rd dimension: density index
c                             4th dimension: parent index
c  input : (r*8) stvim(,,,)= metastable electron impact ionisation
c                            coefficients.
c                             1st dimension: metastable index
c                             2nd dimension: temperature index
c                             3rd dimension: density index
c                             4th dimension: parent index
c  input : (r*8) stvhm(,,,)= metastable charge exchange coefficients
c                             1st dimension: metastable index
```

```
c                              2nd dimension: temperature index
c                              3rd dimension: density index
c  input : (r*8) ratpia(,)    = ratio ( n(z+1)/n(z)  stage abundancies )
c                              1st dimension: temp/dens index
c                              2nd dimension: parent index
c  input : (r*8) ratmia(,)    = ratio ( n(z-1)/n(z)  stage abundancies )
c                              1st dimension: temp/dens index
c                              2nd dimension: parent index
c  input : (r*4) stack(,,,)= population dependence
c                              1st dimension: ordinary level index
c                              2nd dimension: metastable index
c                              3rd dimension: temperature index
c                              4th dimension: density index
c  input : (l*4) lsseta(,) = .true.  - met. ionis rate set in b8gets
c                            .false.- met. ionis rate not set in b8gets
c                              1st dimension: (z) ion metastable index
c                              2nd dimension: (z+1) ion metastable index
c  input : (l*4) lss04a(,) = .true. => ionis. rate set in adf04 file:
c                            .false.=> not set in adf04 file
c                              1st dim: level index
c                              2nd dim: parent metastable index
c
c  input : (i*4) nwvl      = number of wavelength intervals
c  input : (i*4) npix()    = number of pixels in each wvln. interval
c  input : (r*8) wvmin()   = minimum  wvln. (a) for each interval
c  input : (r*8) wvmax()   = maximum  wvln. (a) for each interval
c
c          (r*8) avlt      = lower limit of a-values for pec & f-pec
c
c          (i*4) notrn     = parameter = maximum number of transitions
c          (i*4) ndpec     = parameter = maximum number of pecs per
c                                        metastable for output
c          (i*4) metcnt    = counter of pecs for each metastable
c
c          (i*4) i4unit    = function (see routine selection below)
c
c          (i*4) i         = general use
c          (i*4) j         = general use
c          (i*4) k         = general use
c          (i*4) l         = general use
c
c          (r*8) dum1      = general use- dummy
c          (r*8) dum2      = general use- dummy
c          (r*8) dum3      = general use- dummy
c          (r*8) pec()     = renormalised pec
c                            1st dimension: temperature index
c
c routines:
c          ------------------------------------------------------------
c          hawvrg     adas      check for spectrum line in wvln.interval
c          hapixv     adas      doppler broaden line over pixel range
c          haout1     adas      writes plt and plt-filter output to files
c          b8norm     adas      perform  stage population normalisation
c          b8corp     adas      'fixes' low te problem in rec. data of pecs
c          i4unit     adas      fetch unit number for output of messages
c          xxordr     adas      sorts a real*8 array and its index array
c          xxeiam     adas      return the atomic mass of an element
c          xxmkrc     adas      make root connection vector
c          xxmkrp     adas      make root partition text lines for output
c          xxwcmt_15  adas      writes structured comments to adf15 dataset
c          xxwcmt_40  adas      writes structured comments to adf40 dataset
```

```
c
c author:  h. p. summers, university of strathclyde
c          tel: 0141-548-4196
c
c date:    24/04/02
c
c
c  version  : 1.1
c  date     : 24-02-2003
c  modified : H P Summers
c              - first version.
c
c  version  : 1.2
c  date     : 12-11-2003
c  modified : Martin O'Mullane
c              - trap plt and pltnfl for values below machine precision.
c              - increased number of transitions in line with 801/ifgpp.
c
c  version  : 1.3
c  date     : 05-12-2003
c  modified : Thomas Puetterich
c              - did not write f-pec file as per specification.
c
c  version  : 1.4
c  date     : 25-02-2004
c  modified : Martin O'Mullane
c              - increased number of transitions in line with 801/ifgpp.
c              - change behaviour of plt and filtered plt. no
c                limitations of wavelength or a-value to iunt11. an
c                adf35 filter is now an input and write plt filtered
c                by this to iunt11f.
c
c  version  : 1.5
c  date     : 26-05-2006
c  modified : Hugh Summers
c              - altered output on header lines for superstage
c                 compatibility.
c              - altered strategy for power ranking of emissivities
c              - altered comment lines for superstage compatibility
c                and field key reading of comments.
c
c  version  : 1.6
c  date     : 20-02-2007
c  modified : Martin O'Mullane
c              - Do not write comments to non-open units.
c              - Bring interactive version into line with
c                latest version of adf15 defintion (superstages).
c
c-----------------------------------------------------------------------
      CHARACTER*18        CSTRGA(NDLEV)
      CHARACTER*8         DATE
      CHARACTER*80        DSN35,       DSNEXP,      DSNINC
      CHARACTER*3         TITLED
      CHARACTER*30        USER
      INTEGER             IA(NDLEV),   ICNTE,       ICNTH,       ICNTI
      INTEGER             ICNTR,       IE1A(NDTRN), IE2A(NDTRN)
      INTEGER             IETRN(NDTRN),             IL
      INTEGER             ILA(NDLEV),  IMETR(NDMET)
      INTEGER             IORDR(NDLEV),             ISA(NDLEV),  IUNT11
      INTEGER             IUNT11F,     IUNT15,      IUNT35,      IUNT40
      INTEGER             IZ,          IZ0,         IZ1,         MAXD
```

```
INTEGER             MAXT,         NDDEN,        NDLEV,        NDMET
INTEGER             NDPIX,        NDTEM,        NDTRN,        NDWVL
INTEGER             NMET,         NORD,         NPIX(NDWVL),  NPL
INTEGER             NPL3,         NPLI,         NPLR,         NWVL
LOGICAL             LHSEL,        LIOSEL,       LISEL,        LNORM
LOGICAL             LNSEL,        LPSEL,        LRSEL
LOGICAL             LSS04A(NDLEV,NDMET),        LSSETA(NDMET,NDMET)
LOGICAL             LZSEL,        OPEN11,       OPEN11F,      OPEN15
LOGICAL             OPEN35,       OPEN40
REAL*8              AA(NDTRN),    AVLT,         BWNO
REAL*8              DENSA(NDDEN),               RATMIA(NDDEN,NDMET)
REAL*8              RATPIA(NDDEN,NDMET)
REAL                STACK(NDLEV,NDMET,NDTEM,NDDEN)
REAL*8              STCKM(NDMET,NDTEM,NDDEN)
REAL                STVH(NDLEV,NDTEM,NDDEN,NDMET)
REAL*8              STVHM(NDMET,NDTEM,NDDEN,NDMET)
REAL                STVI(NDLEV,NDTEM,NDDEN,NDMET)
REAL*8              STVIM(NDMET,NDTEM,NDDEN,NDMET)
REAL                STVR(NDLEV,NDTEM,NDDEN,NDMET)
REAL*8              STVRM(NDMET,NDTEM,NDDEN,NDMET)
REAL*8              TEVA(NDTEM),  WA(NDLEV),    WVMAX(NDWVL)
REAL*8              WVMIN(NDWVL),               XJA(NDLEV)
```

8.28 hapixv: Subroutine hapixv from library adas8xx

```
       subroutine hapixv( ndwvl  , ndpix  , fcrit  ,
     &                    nwvl   , npix   , wvmin  , wvmax  ,
     &                    iwvrg  , cpixmx ,
     &                    wvl    , tev    , amssno , pec    ,
     &                    cpixa  , ind1   , ind2
     &                  )

c-----------------------------------------------------------------------
c
c  ****************** fortran77 subroutine: hapixv *********************
c
c  purpose:  distribute Doppler broadened line emission into pixel range
c
c  calling program: hapecf
c
c
c  subroutine:
c
c  input : (i*4)  ndwvl   = maximum number of wavelength intervals
c  input : (i*4)  ndpix   = maximum number of pixels per wvln. interval
c  input : (r*8)  fcrit   = pixel counts for the selected line below
c                           this fraction of the largest pixel count are
c                           discounted.
c
c  input : (i*4)  nwvl    = number of wavelength intervals
c  input : (i*4)  npix()  = number of pixels assigned to wavelength interval
c  input : (r*8)  wvmin() = lower limit of wavelength interval (ang)
c  input : (r*8)  wvmax() = upper limit of wavelength interval (ang)
c
c  input : (i*4)  iwvrg   = index of wavelength range in which line lies
c  input : (r*8)  cpixmx  = largest pixel count currently found
c                           for the wavelength range
c
c  input : (r*8)  wvl     = input line wavelength for test(ang)
c  input : (r*8)  tev     = electron temperature (eV)
c  input : (r*8)  amssno  = atomic mass number
c  input : (r*8)  pec     = photon emissivity coefficient for line
c
c  output: (r*8)  cpixa() = counts in each pixel for the line
c  output: (r*8)  ind1    = first pixel with non-negligible count
c  output: (r*8)  ind2    = last pixel with non-negligible count
c
c routines:
c          routine    source    brief description
c          ------------------------------------------------------------
c          i4unit     adas      fetch unit number for output of messages
c          r8erfc     adas      returns erfc(x) function value
c
c author:  Hugh Summers, University of Strathclyde
c          JA7.08
c          tel. 0141-548-4196
c
c date:    15/01/02
c
c version  : 1.1
c date     : 15-01-2002
c modified : H P Summers
c             - first version.
c
```

```
c version  : 1.2
c date     : 18-06-2007
c modified : H P Summers
c             - corrected error in ind1 & ind2 return.
c
c-----------------------------------------------------------------------
c-----------------------------------------------------------------------
      INTEGER             IND1,         IND2,         IWVRG,        NDPIX
      INTEGER             NDWVL,        NPIX(NDWVL), NWVL
      REAL*8              AMSSNO,       CPIXA(NDPIX),              CPIXMX
      REAL*8              FCRIT,        PEC,          TEV,          WVL
      REAL*8              WVMAX(NDWVL),               WVMIN(NDWVL)
```

8.29 hawvrg: Subroutine hawvrg from library adas8xx

```
C
       subroutine hawvrg( ndwvl  , ndpix  ,
     &                    nwvl   , npix   , wvmin  , wvmax  ,
     &                    wvl    ,
     &                    lwvrg  , iwvrg
     &                  )
c
c-----------------------------------------------------------------------
c
c  ****************** fortran77 subroutine: hawvrg *********************
c
c  purpose:  to check if a line wavelength is in one of the selected
c            wavelength intervals
c
c  calling program: hapecf
c
c
c  subroutine:
c
c  input : (i*4)  ndwvl   = maximum number of wavelength intervals
c  input : (i*4)  ndpix   = maximum number of pixels per wvln. interval
c
c  input : (i*4)  nwvl    = wvaelength intervals
c  input : (i*4)  npix()  = number of pixels assigned to wavelength interval
c  input : (r*8)  wvmin() = lower limit of wavelength interval (ang)
c  input : (r*8)  wvmax() = upper limit of wavelength interval (ang)
c
c  input : (r*8)  wvl     = input line wavelength for test(ang)
c
c  output: (l*4)  lwvrg   = .true.  => spectrum line in selected range
c                         = .false. => spectrum line in selected range
c  output: (i*4)  iwvrg   = index of wavelength range in which lin lies
c                           if lwvrg = .true. otherwise set to zero
c
c routines:
c          routine    source    brief description
c          ------------------------------------------------------------
c          i4unit     adas      fetch unit number for output of messages
c
c author:  Hugh Summers, University of Strathclyde
c          JA7.08
c          tel. 0141-548-4196
c
c date:    15/01/02
c
c update:
c
c-----------------------------------------------------------------------
c-----------------------------------------------------------------------
      INTEGER             IWVRG,       NDPIX,       NDWVL
      INTEGER             NPIX(NDWVL), NWVL
      LOGICAL             LWVRG
      REAL*8              WVL,         WVMAX(NDWVL)
      REAL*8              WVMIN(NDWVL)
```

8.30 rbesf: Subroutine rbesf from library adas8xx

```
C
       FUNCTION RBESF(LAM,Q,X)
       IMPLICIT REAL*8(A-H,O-Z)
C
C  PURPOSE: EVALUATES HALF INTEGER BESSEL FUNCTION
C
C  RBESF=(J(LAM,Q*X)-DELTA(LAM,0))/Q**2
C-----------------------------------------------------------------------
C
C  VERSION  : 1.1
C  DATE     : ?
C  MODIFIED : H P Summers
C              - Initial version.
C
C  VERSION  : 1.2
C  DATE     : 16-05-2007
C  MODIFIED : Allan Whiteford
C              - Remove listing information from colums 72+.
C              - Updated comments as part of subroutine documentation
C          procedure.
C-----------------------------------------------------------------------
       Z=Q*X
       XLAM=LAM
       IF(Z.LE.1.0D0)GO TO 25
       Z0=1.570796*XLAM
       SN=DSIN(Z-Z0)
       CS=DCOS(Z-Z0)
       T=1.0
       RBESF=T*SN
       I=0
       IC=1
    5  I=I+1
       XI=I
       T=T*(XLAM+XI)*(XLAM-XI+1.0)/(XI*2.0*Z)
       IF(DABS(T).LE.1.0D-7)GO TO 20
       GO TO (10,15),IC
   10  RBESF=RBESF+T*CS
       T=-T
       IC=2
       GO TO 5
   15  RBESF=RBESF+T*SN
       IC=1
       GO TO 5
   20  RBESF=RBESF/Z
       IF(LAM.LE.0)RBESF=RBESF-1.0D0
       RBESF=RBESF/(Q*Q)
   60  RETURN
   25  T=1.0
       IF(LAM.LE.0)GOTO 36
       DO 35 I=1,LAM
       XI=I
   35  T=T/(2.0*XI+1.0)
       T=T*X*X
       IF(LAM.NE.2)T=T*Z**(LAM-2)
       I=0
       GO TO 37
   36  T=-(X*X)/6.0D0
       I=1
   37  RBESF=T
```

```
     Z2=0.5*Z*Z
40   I=I+1
     XI=I
     T=-T*Z2/(XI*(2.0*(XLAM+XI)+1.0))
     IF(DABS(T).LE.1.0D-7)GO TO 60
     RBESF=RBESF+T
     GO TO 40
    END
    INTEGER             LAM
    REAL*8              Q,           X
```

8.31 rdwbes: Subroutine rdwbes from library adas8xx

```
C
       subroutine rdwbes( ndinfo,
     &                    z0    , nlqs  , nshell , na    , la    ,
     &                    ea    , qda   , alfaa  ,
     &                    jsn   , jealfa, acc    , xmax  , h     ,
     &                    lam   , einc  , irept  , iext  ,iochk  ,
     &                    res   ,
     &                    ninfo , cinfoa
     &                  )
C-----------------------------------------------------------------------
C
C  *************** fortran77 program: rdwbes.for ********************
C
C  Purpose:  Evaluates Born multipole integrals using distorted bound
C            waves.  The distorted waves are in a Jucys or Slater type
C            potential.
C
C
C  Subroutine:
C
C  Input : (i*4)  ndinfo   = maximum number of information strings
C  Input : (r*8)  z0       = nuclear charge (+ve)
C  Input : (i*4)  nlqs()   = 1000*n+100*l+iqfor each screening shell
C                            1st dim: screening shell index
C  Input : (i*4)  nshell   = number of screening shells
C  Input : (i*4)  na()     = initial (1) and final (2) state principal
C                            quantum numbers.
C  Input : (i*4)  la()     = initial (1) and final (2) orbital quantum
C                            numbers.
C  Input : (r*8)  ea()     = energies(ryd) of initial (1) and final (2)
C                            states - set <0 for bound states.
C  Input : (r*8)  qda()    = quantum defects for initial (1) and
C                            final (2) states.
C  Input : (r*8)  alfaa(,) = screening parameters
C                            1st dim: initial (1) and final (2) states
C                            2nd dim: screening shell index.
C  Input : (i*4)  jsn      = -1 => Jucys potential
C                          =  0 => Slater potential
C  Input : (i*4)  jealfa   =  0 => search for energies given potential
C                          =  1 => search for alfaa parameters for
C                                  potential given energies and quantum
C                                  defects.
C  Input : (r*8)  acc      = search accuracy setting
C  Input : (r*8)  xmax     = range for numerical wave function generation
C                            and storage
C  Input : (r*8)  h        = step interval for numerical wave function
C                            storage
C  Input : (i*4)  lam      = Born multipole
C  Input : (r*8)  einc     = incident electron energy(ryd)
C  Input : (i*4)  irept    = 0 =>full wave function determination
C                          = 1 => repeat with same wave functions as in
C                                 previous case with irept=0
C  Input : (i*4)  iext     = 0 normal operation with internally generated
C                            wave functions
C                          = 1 use external wave functions supplied in
C                            function gext(x,n,l) with n and l
C                            specifying orbital.
C  Input : (i*4)  iochk    = 1 => Born multipole integral evaluated
C                          = 2 => Ochkur multipole integral evaluated
```

```
C
C  Output: (r*8)  res      = multipole integral (at.unit)
C  Output: (i*4)  ninfo    = number of information strings
C  Output  (c*90) cinfoa() = information strings
C                            1st dim: index number of strings
C
C
C  Routines:
C          routine    source   brief description
C          ------------------------------------------------------------
C          zeff       adas      effectrive charge (+ve)
C          effz3      adas      evaluates effective potential
C          gext       adas      access external radial wave functions
C          zser       adas      power series expansion of z(r)
C          bdcf7      adas      generate bound radial distorted wave fn.
C          fcf6       adas      generate free radial distorted wave fn.
C          rbesf      adas      evalluate bessel function
C          ass2       adas      asymptotic integral contribution
C          dnamp      adas      asymp. wave fn. amplitude Taylor coeffts
C          phase      adas      asymp. wave fn. phase Taylor coeffts
C          ass        adas      asymptotic integral contribution
C
C  Author:  H. P. Summers, University of Strathclyde
C           ja7.08
C           tel. 0141-548-4196
C
C  Date:   22/04/85
C
C  Update: HP Summers  16/06/95  alter definition of nlqs as
C                                1000*n+100*l+iq to avoid problem when
C                                number of equivalent electrons is 10
C  Update: HP Summers  21/05/04  restructure and add calculation
C                                information strings to parameter output
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      CHARACTER*90        CINFOA(NDINFO)
      INTEGER             IEXT,        IOCHK,       IREPT,       JEALFA
      INTEGER             JSN,         LA(2),       LAM,         NA(2)
      INTEGER             NDINFO,      NINFO,       NLQS(10),    NSHELL
      REAL*8              ACC,         ALFAA(2,10), EA(2),       EINC
      REAL*8              H,           QDA(2),      RES,         XMAX
      REAL*8              Z0
```

9 Subroutine library adaslib

9.1 argam: Subroutine argam from library adaslib

```
C
       function argam( l   , a   )
c-----------------------------------------------------------------------
c
c  ****************** fortran77 function: argam   **********************
c
c  purpose:  calculates arggamma(l+1+i*a) where l is an integer not
c            less than zero
c
c
c  subroutine:
c
c  input : (i*4)  l         = angula momentum >=0
c  input : (r*8)  a         = real argument
c
c  output: (r*8)  argam     = arggamma(l+1+i*a)
c
c  routines:
c          none
c
c  author:  h. p. summers, university of strathclyde
c           ja7.08
c           tel. 0141-548-4196
c
c  date:    06/06/02
c
c  update:
c
C VERSION: 1.1                          DATE: 04-07-96
C MODIFIED: WILLIAM OSBORN
C               - FIRST VERSION.
C VERSION: 1.2                          DATE: 14-06-2000
C MODIFIED: Martin O'Mullane
C               - changed definition from real argam*8().
C
C VERSION: 1.3                          DATE: 19-12-01
C MODIFIED: Martin O'MULLANE
C               - Removed junk from > column 72.
C
C VERSION: 1.4                          DATE: 18-03-03
C MODIFIED: Hugh Summers
C               - Re-written and updated.
c
c-----------------------------------------------------------------------
c-----------------------------------------------------------------------
      INTEGER             L
      REAL*8              A
```

9.2 bf: Subroutine bf from library adaslib

```
       SUBROUTINE BF(G,N,A)
C
       IMPLICIT REAL*8 (A-H,O-Z)
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: BF *************************
C
C PURPOSE: Evaluates the hydrogenic <nl|r|kl> bound-free radial integral
C
C NOTES: THIS ROUTINE IS NOT YET PROPERLY ANNOTATED
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 16-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST VERSION
C
C VERSION  : 1.2
C DATE     : 10-04-2007
C MODIFIED : Allan Whiteford
C               - Modified documentation as part of automated
C   subroutine documentation preparation.
C
C-----------------------------------------------------------------------
C
      INTEGER             N
      REAL*8              A,           G(1000)
```

9.3 ceigrp: Subroutine ceigrp from library adaslib

```
      FUNCTION CEIGRP(CHR)
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 FUNCTION: CEIGRP  ********************
C
C  PURPOSE: RETURNS EISSNER CODE FOR AN ORBITAL
C
C
C  INPUT :  (C*2)   CHR    = ORBITAL (EG.  '2S')
C
C  OUTPUT:  (C*1)   CEIGRP = EISSNER CODE
C
C           (I*4)   I      = GENERAL INDEX (EG. 'A')
C
C  SUBROUTINES:  NONE
C  AUTHOR:  W.J. DICKSON, JET
C
C  DATE:    20/10/89

C
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 22-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - PUT UNDER SCCS CONTROL
C VERSION: 1.2                          DATE: 19-02-03
C MODIFIED:  H.P. SUMMERS
C - EXTENDED RANGE
C           - ALTERED ORBITALS TO INCLUDE 'J'
C           - A POINT OF CONFUSION BETWEEN CONDON
C              & SHORTLEY, COWAN AND ADASDR
C
C-----------------------------------------------------------------------
      CHARACTER*2         CHR
```

9.4 ceprep: Subroutine ceprep from library adaslib

```
C
      FUNCTION ceprep(str)
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN 77 FUNCTION: CEPREP ********************
C
C  PURPOSE: Prepares a string for passsing to xxdtes. Takes C*18
C           and returns C*19; add a space to start if not d10 or
C           f10-f14. In this case a space is added at the end.
C
C  CALLING PROGRAM: VARIOUS
C
C  SUBROUTINE:
C
C  INPUT  :  (C*18)   str    = input configuration (EISSNER)
C  OUTPUT :  (C(19)   ceprep = lengthened configuration
C
C
C  ROUTINES: NONE
C
C  AUTHOR   :  Martin O'Mullane
C  DATE     :  21-11-2003
C
C
C  VERSION  : 1.1
C  DATE     :  21-11-2003
C  MODIFIED : Martin O'Mullane
C             - First version.
C
C  VERSION  : 1.2
C  DATE     :  15-11-2004
C  MODIFIED : Martin O'Mullane
C             - Do not change configuration if first character is a space.
C
C-----------------------------------------------------------------------------
      CHARACTER*18        STR
```

9.5 check_pipe: Subroutine check_pipe from library adaslib

```
       SUBROUTINE CHECK_PIPE
C-----------------------------------------------------------------------
C
C  ******************** FORTRAN77 PROGRAM: CHECK_PIPE **********************
C
C  VERSION:  1.0
C
C  PURPOSE:  DIAGNOSTIC CHECK ON THE STATE OF THE FORTRAN-IDL PIPE
C
C  NOTES:  THE EQUIVALENT IDL CHECK_PIPE PROCEDURE MUST BE PUT IN
C          THE IDL CODE TO COMMUNICATE WITH THIS FORTRAN SUBROUTINE.
C          BOTH ROUTINES WRITE A STRING, READ A STRING AND THEN CHECK
C          WHETHER THAT STRING IS AS EXPECTED. A MESSAGE IS THEN
C          OUTPUT INDICATING WHETHER THE PIPE WAS EMPTY BEFORE THE CALLS
C          WERE MADE, SO TELLING THE USER WHETHER PIPE COMMUNICATIONS
C          BETWEEN THE IDL AND FORTRAN HAD BEEN SUCCESSFUL UP TO THAT
C          POINT.
C
C          RATHER THAN GIVING THE 'ERROR' MESSAGE, THE FORTRAN MAY CRASH
C          INDICATING THAT THE IDL HAS WRITTEN TOO MUCH DATA TO THE PIPE
C          BEFORE THIS CALL AND THAT THE DATA WAS NOT A STRING.
C
C          IF THE FORTRAN OR IDL HANGS IN THIS ROUTINE THEN THERE IS A
C          MAJOR (>129 CHARACTER) BLOCKAGE IN THE PIPE.
C
C  PROGRAM:
C     (I*4)  PIPEIN    =  PIPE INPUT STREAM FROM IDL
C     (I*4)  PIPEOU    =  PIPE OUTPUT STREAM TO IDL
C     (C*129) A        =  OUTPUT STRING
C     (C*129) B        =  INPUT STRING
C
C  ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          XXFLSH     ADAS      FLUSHES I/O BUFFER
C          I4UNIT     ADAS      STANDARD ERROR OUTPUT (SET UP IN XX0000)
C
C-----------------------------------------------------------------------
C AUTHOR:  WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C
C DATE:    5TH JULY 1996
C
C VERSION: 1.1                          DATE: 05-07-96
C MODIFIED: WILLIAM OSBORN
C               - FIRST VERSION.
C-----------------------------------------------------------------------
C
```

9.6 continuo: Subroutine continuo from library adaslib

```
      subroutine continuo(wave   , tev    , iz0  , iz1  ,
     &                    contff , contin
     &                   )

C-----------------------------------------------------------------------
C
C  ***************** FORTRAN77 SUBROUTINE: continuo *******************
C
C  PURPOSE: For an given wavelength generate radiative reconbination
C           and bremsstrahlung emissivity.
C
C  CALLING PROGRAM: General use.
C
C
C  INPUT : (R*8)  wave    = Wavelength required (A)
C          (R*8)  tev     = Electron temperature (eV)
C          (I*4)  iz0     = Atomic number
C          (I*4)  iz1     = Ion stage + 1
C  OUTPUT: (R*8)  contff  = Free-free emissivity (ph cm3 s-1 A-1)
C          (R*8)  contin  = Total continuum emissivity
C                           (free-free + free-bound)(ph cm3 s-1 A-1)
C
C
C  ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          r8gbf      ADAS      Free-free Gaunt factor.
C          r8gav      ADAS      Total gaunt factor for free-free and
C                               quasi-continuous free-bound transitions
C          r8giiiav   ADAS      Maxwellian averaged free-free Gaunt factors
C
C
C  NOTES :  Based on Lorne Horton's original continuo program but
C           coefficients, rather than emission, are returned.
C
C
C
C  AUTHOR:  Martin O'Mullane
C  DATE  :  02-03-2005
C
C
C
C  VERSION : 1.1
C  DATE    : 02-03-2005
C  MODIFIED: Martin O'Mullane
C            - First version
C
C  VERSION : 1.2
C  DATE    : 02-03-2005
C  MODIFIED: Martin O'Mullane
C            - Alter comments to note that contin is the sum of
C              free-free and free-bound continuum emission.
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IZ0,         IZ1
      REAL*8              CONTFF,      CONTIN,      TEV,         WAVE
```

9.7 cstgrp: Subroutine cstgrp from library adaslib

```
      FUNCTION CSTGRP(CHR)
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 FUNCTION: CSTGRP  ********************
C
C  PURPOSE: RETURNS TERM OF ORBITAL GIVEN IN THE EISSNER SINGLE
C           HEXADECIMAL CHARACTER FORM
C
C
C  INPUT :  (C*1)   CHR    = HEX. ORBITAL CHARACTER (EG. 'A')
C
C  OUTPUT:  (C*2)   CSTGRP = ORBITAL TERM   (E.G. '4F')
C
C           (I*4)   I      = GENERAL INDEX
C
C  SUBROUTINES:  NONE
C  AUTHOR:  W.J. DICKSON, JET
C
C  DATE:    20/10/89
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 22-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - PUT UNDER SCCS CONTROL
C
C VERSION: 1.2                          DATE: 19-02-03
C MODIFIED: H.P. SUMMERS
C - EXTENDED RANGE
C                 - ALTERED ORBITALS TO INCLUDE 'J'
C                 - A POINT OF CONFUSION BETWEEN CONDON
C                   & SHORTLEY, COWAN AND ADASDR
C
C-----------------------------------------------------------------------
      CHARACTER           CHR
```

9.8 ee2: Subroutine ee2 from library adaslib

```
       FUNCTION EE2(X)
C
C-----------------------------------------------------------------------
C
C  ***************** FORTRAN77 FUNCTION: EE2 *************************
C
C PURPOSE: EVALUATES EXP(X)E2(X) WHERE E2 IS THE 2ND EXPONENTIAL
C          INTEGRAL
C
C CALLING PROGRAMS: GENERAL
C
C INPUT:  (R*8)  X       = INDEPENDENT VARIABLE
C
C OUTPUT: (R*8)  EE2     = EXP(X)E2(X)
C
C ROUTINES:
C          ROUTINE    SOURCE   BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          EEI        ADAS     EVALUATES 1ST EXPONENTIAL INTEGRAL
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 11-07-95
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - PUT UNDER S.C.C.S. CONTROL
C
C VERSION: 1.2                          DATE: 06-03-96
C MODIFIED: TIM HAMMOND
C               - ADDED HEADERS
C
C VERSION: 1.3                          DATE: 17-4-07
C MODIFIED: HUGH SUMMERS
C           - COMPLETED COMMENT BLOCK DESCRIPTION
C
C-----------------------------------------------------------------------
C
       IMPLICIT REAL*8(A-H,O-Z)
       IF(X-30.0D0)1,1,2
    1  EE2=1.0D0-X*EEI(X)
       GO TO 3
    2  X1=1.0D0/X
       EE2=X1*(1.0D0-X1*(2.0D0-X1*(6.0D0-X1*(24.0D0-X1*(120.0D0-X1*
     1(720.0D0-X1*5040.0D0))))))
    3  RETURN
      END
      REAL*8              X
```

9.9 ee3: Subroutine ee3 from library adaslib

```
       FUNCTION EE3(X)
C
       IMPLICIT REAL*8(A-H,O-Z)
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 FUNCTION: EE3 *************************
C
C PURPOSE: EVALUATES EXP(X)E3(X) WHERE E3 IS THE 1ST EXPONENTIAL
C          INTEGRAL
C
C CALLING PROGRAMS: GENERAL
C
C INPUT:  (R*8)  X       = INDEPENDENT VARIABLE
C
C OUTPUT: (R*8)  EE3     = EXP(X)E3(X)
C
C ROUTINES:
C          ROUTINE    SOURCE   BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          EE2        ADAS     EVALUATES 2ND EXPONENTIAL INTEGRAL
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 06-03-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - PUT UNDER S.C.C.S. CONTROL
C
C VERSION: 1.2                          DATE: 17-4-07
C MODIFIED: HUGH SUMMERS
C           - COMPLETED COMMENT BLOCK DESCRIPTION
C
C-----------------------------------------------------------------------
C
       IF(X-30.0D0)1,1,2
    1  EE3=0.5D0*(1.0D0-X*EE2(X))
       GO TO 3
    2  X1=1.0D0/X
       EE3=X1*(1.0D0-X1*(3.0D0-X1*(12.0D0-X1*(60.0D0-X1*(360.0D0-X1*
     1(2520.0D0-X1*20160.0D0))))))
    3  RETURN
      END
      REAL*8             X
```

9.10 eei: Subroutine eei from library adaslib

```
      FUNCTION EEI(X)
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 FUNCTION: EEI *************************
C
C PURPOSE: EVALUATES EXP(X)E1(X) WHERE E1 IS THE 1ST EXPONENTIAL
C          INTEGRAL
C
C CALLING PROGRAMS: GENERAL
C
C INPUT:  (R*8)  X       = INDEPENDENT VARIABLE
C
C OUTPUT: (R*8)  EEI     = EXP(X)E1(X)
C
C ROUTINES: NONE
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 11-7-95
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST VERSION
C
C VERSION: 1.2                          DATE: 16-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - TIDIED UP COMMENTS
C
C VERSION: 1.3                          DATE: 17-4-07
C MODIFIED: HUGH SUMMERS
C           - COMPLETED COMMENT BLOCK DESCRIPTION
C
C-----------------------------------------------------------------------
C
      IF (X.LE.1.0D0) THEN
C
         A = -LOG(X) - 0.57721566D0 + X * (0.99999193D0 - X *
     +       (0.24991055D0 - X * (0.05519968D0 - X *
     +       (0.00976004D0 - X * 0.00107857D0))))
         Y = 0.5D0 * X
         Z = 1.0D0 - Y * (0.9998684D0 - Y * (0.4982926D0 - Y *
     +       (0.1595332D0 - Y * 0.0293641D0)))
         EEI = A / (Z * Z)
C
      ELSE
C
         EEI = (0.2677737343D0 + X * (8.6347608925D0 + X *
     +         (18.059016973D0 + X * (8.5733287401D0 + X)))) /
     +         (X * (3.9584969228D0 + X * (21.0996530827D0 + X *
     +         (25.6329561486D0 + X * (9.5733223454D0 + X)))))
C
      ENDIF
C
      RETURN
      END
      REAL*8             X
```

9.11 i4eiss: Subroutine i4eiss from library adaslib

```
      FUNCTION I4EISS( CHR )
C-----------------------------------------------------------------------
C                                                                 ********
C  ****************** FORTRAN 77 FUNCTION: I4EISS ******************  ********
C
C  PURPOSE: RETURNS DECIMAL INDEX OF AN ORBITAL GIVEN IN THE EISSNER
C           HEXADECIMAL CHARACTER FORM.
C
C  CALLING PROGRAM: VARIOUS
C
C  SUBROUTINE:
C
C  INPUT  :  (C*1)   CHR    = HEX. ORBITAL CHARACTER (EISSNER)
C  OUTPUT :  (I*4)   I4EISS = DECIMAL ORBITAL INDEX (1 - 61)
C
C            (I*4)   I      = GENERAL INDEX
C            (C*1)   CHRA() = EISSNER HEXADECIMAL ORBITAL LIST
C
C  ROUTINES: NONE
C
C  AUTHOR:  H. P. SUMMERS, UNIVERSITY OF STRATHCLYDE
C           JA8.08
C           TEL. 0141-553-4196
C
C  DATE:    04/06/96
C
C UNIX-IDL PORT:
C    WILLIAM OSBORN, TESSELLA SUPPORT SERVICES PLC.
C
C DATE:    20TH AUGUST 1996
C
C VERSION: 1.1 DATE: 20-08-96
C MODIFIED: WILLIAM OSBORN
C     - FIRST VERSION. SET DEFAULT VALUE OF I4EISS TO ZERO
C VERSION: 1.2 DATE: 19-02-03
C MODIFIED: H.P. SUMMERS - EXTENDED RANGE
C-----------------------------------------------------------------------
      CHARACTER           CHR
```

9.12 i4eiz0: Subroutine i4eiz0 from library adaslib

```
      FUNCTION I4EIZ0 ( ESYM )
C-----------------------------------------------------------------------
C
C *************** FORTRAN77 INTEGER*4 FUNCTION: I4EIZ0 *****************
C
C PURPOSE: TO RETURN THE NUCLEAR CHARGE FOR THE ELEMENT SYMBOL ESYM
C          (INTEGER*4 FUNCTION VERSION OF 'XXEIZ0')
C
C CALLING PROGRAM: GENERAL USE
C
C FUNCTION:
C
C          (I*4)  I4EIZ0  = FUNCTION NAME -
C                           ELEMENT NUCLEAR CHARGE
C          (C*2)  ESYM    = SYMBOL OF ELEMENT WITH NUCLEAR CHARGE I4EIZ0
C
C          (I*4)  NSYM    = PARAMETER = NUMBER OF SYMBOLS LISTED
C
C          (I*4)  I       = GENERAL ARRAY USE
C
C          (C*2)  SYMBOL()= SYMBOLS OF FIRST 'NSYM' ELEMENTS (NORMAL).
C                           ARRAY DIMENSION => NUCLEAR CHARGE
C          (C*2)  SYMBLC()= SYMBOLS OF FIRST 'NSYM' ELEMENTS (L.C.).
C                           ARRAY DIMENSION => NUCLEAR CHARGE
C          (C*2)  SYMBUC()= SYMBOLS OF FIRST 'NSYM' ELEMENTS (U.C.).
C                           ARRAY DIMENSION => NUCLEAR CHARGE
C
C NOTES:    IF SYMBOL IS NOT RECOGNISED, I.E.NOT IN Z0 RANGE 1 & 'NSYM',
C           THEN THE INTEGER 'I4EIZ0' IS RETURNED AS ZERO.
C
C ROUTINES: NONE
C
C
C AUTHOR:   PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C           K1/0/81
C           JET EXT. 4569
C
C DATE:     13/02/91
C
C UPDATE:
C
C VERSION 1.2:   DATE: 30/01/98
C MODIFIED: HP SUMMERS
C - ALLOWED SEQUENCE SYMBOL TO BE IN  UPPER, LOWER OR MIXED
C   CASE.
C VERSION 1.3:   DATE: 37/09/99
C MODIFIED: HP SUMMERS
C - EXTENDED ELEMENT RANGE TO URANIUM.
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      CHARACTER*2         ESYM
```

9.13 i4fctn: Subroutine i4fctn from library adaslib

```
      FUNCTION I4FCTN( STR , IABT )
C-----------------------------------------------------------------------
C
C  *************** FORTRAN77 INTEGER*4 FUNCTION: I4FCTN ****************
C
C  PURPOSE:  TO CONVERT AN INTEGER NUMBER STORED IN A STRING
C            INTO A INTEGER*4 VARIABLE
C
C  CALLING PROGRAM: GENERAL USE
C
C  FUNCTION:
C
C          (I*4)   I4FCTN  = FUNCTION NAME
C          (C*(*)) STR     = STRING CONTAINING SINGLE FLOATING POINT NO.
C          (I*4)   IABT    = RETURN CODE:
C                                0 => NO ERROR
C                                1 => ERROR (A VALUE 'I4FCTN=0' WILL BE
C                                           RETURNED).
C
C          (C*1)   CH0      = PARAMETER = '0'
C          (C*1)   CH9      = PARAMETER = '9'
C          (C*1)   BLANK    = PARAMETER = ' '
C          (C*1)   CPLUS    = PARAMETER = '+'
C          (C*1)   CMINUS   = PARAMETER = '-'
C
C          (I*4)   ILEN     = LENGTH OF 'STR' STRING IN BYTES
C          (I*4)   ILAST    = POSITION OF LAST BYTE OF IDENTIFIED NUMBER
C          (I*4)   I1       = STARTING BYTE IN 'STR' OF NUMBER
C                             INCLUDING SIGN IF PRESENT
C          (I*4)   IS       = 0 => NUMBER HAS NO SIGN
C                             1 => NUMBER HAS A SIGN
C          (I*4)   ICH0     = ICHAR('0')
C          (I*4)   ICH9     = ICHAR('9')
C          (I*4)   ISTR     = ICHAR(CURRENT BYTE POSITION IN 'STR')
C          (I*4)   I        = GENERAL USE
C
C          (L*4)   LFOUND   = .TRUE.  => ALL OF THE INPUT NUMBER BYTES
C                                        HAVE BEEN ASSESSED.
C                             .FALSE. => INPUT NUMBER BYTES STILL BEING
C                                        ASSESSED.
C          (L*4)   LSTART   = .TRUE.  => THE FIRST DIGIT HAS BEEN FOUND
C                             .FALSE. => THE FIRST DIGIT HAS NOT YET
C                                        BEEN REACHED.
C
C          (C*5)   CFORM5   = FORMAT FOR INTERNAL READING OF INTEGER
C
C
C NOTE:     AN ERROR WILL OCCUR (IABT=1) IF THERE IS MORE THAN ONE
C           NUMBER OCCURING IN THE STRING 'STR()'
C
C
C AUTHOR:   PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C           K1/0/37
C           JET EXT. 2520
C
C DATE:     11/07/90
C
C UPDATE:   11/02/92 - PE BRIDEN: BLANKS NOW ALLOWED BETWEEN SIGN AND
C                                 FIRST DIGIT. LSTART VARIABLE ADDED.
```

```
C                                VARIABLE I2 REMOVED.
C                                + SOME MINOR RECODING - (IF STRING
C                                ENTERED IS BLANK IABT IS NOW SET TO 1)
C
C UPDATE:   16/08/93 - PE BRIDEN: CORRECTED BUG TO ALLOW BLANKS BETWEEN
C                                SIGN AND FIRST DIGIT (SEE ABOVE).
C                                1) ILAST VARIABLE ADDED.
C                                2) FORMATTED READ USED INSTEAD OF *
C                                   WHEN CONVERTING IDENTIFIED INTEGER
C                                   USING THE INTERNAL READ. (THIS
C                                   RESTRICTS IDENTIFIED NUMBER TO BE
C                                   < 100 BYTES IN LENGTH!)
C                                3) EXCLUDE TRAILING BLANKS IN THE
C                                   INTERNAL READING OF THE INTEGER
C                                   I.E. STR(I1:ILAST) INSTEAD OF
C                                        STR(I1:ILEN)
C
C UPDATE:   07/03/95 - PE BRIDEN: INSTEAD OF USING FORMAT SPECIFIER I99
C                                WHEN INTERNALLY READING THE INTEGER
C                                CREATE THE APPROPRIATE SPECIFIER
C                                WITHIN CFORM5 AND USE THIS.
C
C VERSION  : 1.3
C DATE     : 20-12-2001
C MODIFIED : Martin O'Mullane
C               - Removed mainframe listing information beyond column 72.
C
C VERSION  : 1.3
C DATE     : 10-04-2007
C MODIFIED : Allan Whiteford
C               - Modified documentation as part of automated
C   subroutine documentation preparation.
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      CHARACTER*(*)       STR
      INTEGER             IABT
```

9.14 i4idfl: Subroutine i4idfl from library adaslib

```
      FUNCTION I4IDFL( N , L )
C-----------------------------------------------------------------------
C
C  *************** FORTRAN77 INTEGER*4 FUNCTION: I4INDL ****************
C
C  PURPOSE:  RETURNS A UNIQUE INDEX NUMBER BASED ON THE VALUE OF THE
C            N AND L QUANTUM NUMBERS PASSED TO IT. THE INDEX IS USED TO
C            REFERENCE ARRAYS CONTAINING DATA DEPENDENT ON THE N AND L
C            QUANTUM NUMBERS.
C
C  CALLING PROGRAM: GENERAL USE
C
C  FUNCTION:
C
C  FUNC:   (I*4)   I4IDFL  = INDEX
C
C  INPUT:  (I*4)   N       = N QUANTUM NUMBER.
C  INPUT:  (I*4)   L       = L QUANTUM NUMBER.
C
C AUTHOR:   JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C           K1/0/81
C           JET EXT. 5183
C
C DATE:     10/09/93
C
C VERSION  : 1.2
C DATE     : 20-12-2001
C MODIFIED : Martin O'Mullane
C               - Removed mainframe listing information beyond column 72.
C
C VERSION  : 1.3
C DATE     : 10-04-2007
C MODIFIED : Allan Whiteford
C               - Modified documentation as part of automated
C   subroutine documentation preparation.
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      INTEGER             L,           N
```

9.15 i4idfm: Subroutine i4idfm from library adaslib

```
      FUNCTION I4IDFM( N , L , M )
C-----------------------------------------------------------------------
C
C  *************** FORTRAN77 INTEGER*4 FUNCTION: I4INDL ***************
C
C  PURPOSE:  RETURNS A UNIQUE INDEX NUMBER BASED ON THE VALUE OF THE
C            N, L AND M QUANTUM NUMBERS PASSED TO IT. THE INDEX IS USED
C            TO REFERENCE ARRAYS CONTAINING DATA DEPENDENT ON THE
C            N, L AND M QUANTUM NUMBERS.
C
C  CALLING PROGRAM: GENERAL USE
C
C  FUNCTION:
C
C  FUNC:   (I*4)   I4IDFM  = INDEX
C
C  INPUT:  (I*4)   N       = N QUANTUM NUMBER.
C  INPUT:  (I*4)   L       = L QUANTUM NUMBER.
C  INPUT:  (I*4)   M       = M QUANTUM NUMBER.
C
C AUTHOR:   JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C           K1/0/81
C           JET EXT. 5183
C
C DATE:     10/09/93
C
C VERSION  : 1.2
C DATE     : 20-12-2001
C MODIFIED : Martin O'Mullane
C               - Removed mainframe listing information beyond column 72.
C
C VERSION  : 1.3
C DATE     : 10-04-2007
C MODIFIED : Allan Whiteford
C               - Modified documentation as part of automated
C   subroutine documentation preparation.
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      INTEGER             L,           M,           N
```

9.16 i4idli: Subroutine i4idli from library adaslib

```
      FUNCTION I4IDLI( NF , LI , LF )
C
C
C-----------------------------------------------------------------------
C
C  *************** FORTRAN77 INTEGER*4 FUNCTION: I4IDLI ****************
C
C  PURPOSE:  RETURNS THE INDEX NUMBER OF THE PREDICTED SPETRUM LINE
C            TABLES GIVEN THE ORBITAL QUANTUM NUMBER OF THE INTIAL STATE
C            AND THE PRINCIPAL AND ORBITAL QUANTUM NUMBERS OF THE FINAL
C            STATE.
C
C  CALLING PROGRAM: GENERAL USE
C
C  FUNCTION:
C
C  FUNC:   (I*4)   I4IDLI  = INDEX
C
C  INPUT:  (I*4)   NF      = PRINCIAPL QUANTUM NUMBER OF FINAL STATE.
C  INPUT:  (I*4)   LI      = ORBITAL QUANTUM NUMBER OF INITIAL STATE.
C  INPUT:  (I*4)   LF      = ORBITAL QUANTUM NUMBER OF FINAL STATE.
C
C AUTHOR:  JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 5183
C
C DATE:    14/10/93
C
C VERSION  : 1.2
C DATE     : 10-04-2007
C MODIFIED : Allan Whiteford
C               - Modified documentation as part of automated
C   subroutine documentation preparation.
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      INTEGER             LF,          LI,          NF
```

9.17 i4indf: Subroutine i4indf from library adaslib

```
C
      FUNCTION I4INDF( ndim , array ,val )


C-----------------------------------------------------------------------
C
C  *************** FORTRAN77 INTEGER*4 FUNCTION: I4INDF ****************
C
C  PURPOSE:  Finds the index in array corresponding to the closest match
C            to value.
C
C  CALLING PROGRAM: GENERAL USE
C
C  FUNCTION:
C
C  FUNC:   (I*4)   I4INDF  = INDEX  (if the entry is outside the range
C                                    of array then -1 is returned.)
C
C  INPUT:  (I*4)   NDIM    = SIZE OF ARRAY
C  INPUT:  (R*8)   ARRAY   = obvious
C  INPUT:  (R*8)   VAL     = SOUGHT VALUE
C
C
C  AUTHOR   : Martin O'Mullane,
C             K1/1/43,
C             JET
C
C  VERSION  : 1.1
C  DATE     : 17/03/1999
C
C  MODIFIED : Martin O'Mullane
C             First version.
C
C VERSION  : 1.2
C DATE     : 10-04-2007
C MODIFIED : Allan Whiteford
C               - Modified documentation as part of automated
C   subroutine documentation preparation.
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      INTEGER             NDIM
      REAL*8              ARRAY(NDIM), VAL
```

9.18 i4indfi4: Subroutine i4indfi4 from library adaslib

```
C
      FUNCTION I4INDFI4( ndim , array , ind )


C-----------------------------------------------------------------------
C
C  ************** FORTRAN77 INTEGER*4 FUNCTION: I4INDFI4 ***************
C
C  PURPOSE : Finds the index in the integer array corresponding to the
C            exact match to value. If the entry is outside the range of
C            array then -1 is returned.
C
C  CALLING PROGRAM: GENERAL USE
C
C  FUNCTION:
C
C  FUNC:   (I*4)   I4INDFI4  = INDEX
C
C  INPUT:  (I*4)   NDIM      = SIZE OF ARRAY
C  INPUT:  (R*8)   ARRAY     = obvious
C  INPUT:  (R*8)   IND       = SOUGHT VALUE
C
C
C  AUTHOR   : Martin O'Mullane,
C
C  VERSION  : 1.1
C  DATE     : 15-04-2005
C
C  MODIFIED : Martin O'Mullane
C               - First version.
C
C VERSION  : 1.2
C DATE     : 10-04-2007
C MODIFIED : Allan Whiteford
C               - Modified documentation as part of automated
C   subroutine documentation preparation.
C
C VERSION  : 1.3
C DATE     : 21-07-2008
C MODIFIED : Martin O'Mullane
C               - Do not assume a monotonic array when checking limits,
C                 determine min/max index values instead.
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      INTEGER             ARRAY(NDIM), IND,         NDIM
```

9.19 i4indfvs: Subroutine i4indfvs from library adaslib

```
      FUNCTION I4INDFVS( ndim , array ,val )


C-----------------------------------------------------------------------
C
C  *************** FORTRAN77 INTEGER*4 FUNCTION: I4INDF ****************
C
C  PURPOSE:  Finds the index in array corresponding to the closest match
C            to value.
C
C  NOTES:    This is a version of i4indf but array need not be
C            monotonically increasing. (vs - variable spacing).
C
C  CALLING PROGRAM: GENERAL USE
C
C  FUNCTION:
C
C  FUNC:   (I*4)   I4INDFVS  = INDEX
C
C  INPUT:  (I*4)   NDIM      = SIZE OF ARRAY
C  INPUT:  (R*8)   ARRAY     = obvious
C  INPUT:  (R*8)   VAL       = SOUGHT VALUE
C
C
C  AUTHOR   : Martin O'Mullane,
C
C  VERSION  : 1.1
C  DATE     : 23-07-2003
C  MODIFIED : Martin O'Mullane
C               - First version.
C
C VERSION  : 1.2
C DATE     : 10-104-2007
C MODIFIED : Allan Whiteford
C                - Modified documentation as part of automated
C   subroutine documentation preparation.
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      INTEGER             NDIM
      REAL*8              ARRAY(NDIM), VAL
```

9.20 i4jgam: Subroutine i4jgam from library adaslib

```
      FUNCTION I4JGAM( INDEX )
C
C
C-----------------------------------------------------------------------
C
C  ***************** FORTRAN77 INTEGER FUNCTION: I4JGAM *****************
C
C  PURPOSE: USES INDEX TO REFERENCE 'JGAM' TABLE GENERATED BY SUBROUTINE
C           'XXGAMA'.
C
C           THE 'GAM' TABLE IS REFERENCED WITH THE FUNCTION 'R8GAM'.
C
C
C  CALLING PROGRAM: GENERAL USE.
C
C  FUNC  : (I*4)  I4JGAM  =
C
C  INPUT : (I*4)  INDEX   =
C
C  PARAM : (I*4)  MXINDX  = 200
C
C          (L*4)  LFIRST   = .TRUE.  = FIRST TIME FUNCTION CALLED.
C                            .FLASE. = FUNCTION HAS BEEN CALLED BEFORE.
C
C          (I*4)  JGAM()  =
C                           DIMENSION: REFERENCED BY 'INDEX'.
C          (I*4)  GAM()   =
C                           DIMENSION: REFERENCED BY 'INDEX'.
C
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4UNIT     ADAS      RETURNS UNIT NO. FOR OUPUT OF MESSAGES.
C          XXGAMA     ADAS      FILLS 'JGAM'.
C
C AUTHOR:   JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C           K1/0/81
C           JET EXT. 5183
C
C DATE:     30/09/93
C
C VERSION  : 1.2
C DATE     : 20-12-2001
C MODIFIED : Martin O'Mullane
C               - Removed mainframe listing information beyond column 72.
C
C VERSION  : 1.3
C DATE     : 10-04-2007
C MODIFIED : Allan Whiteford
C               - Modified documentation as part of automated
C   subroutine documentation preparation.
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             INDEX
```

9.21 i4jgrp: Subroutine i4jgrp from library adaslib

```
      INTEGER FUNCTION I4JGRP(CHR)
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 FUNCTION: I4JGRP  ********************
C
C  PURPOSE: RETURNS DECIMAL FORM OF EISSNER SINGLE HEX CHARACTER
C           ORBITAL GIVEN IN THE EISSNER SINGLE HEXADECIMAL CHARACTER
C           FORM
C
C  INPUT  (C*2)   CHR    = HEX. ORBITAL CHARACTER (EG. 'A')
C
C  OUTPUT  (I*4)  I4JGRP = DECIMAL ORBITAL CHARACTER (E.G. '10')
C
C  SUBROUTINES:  NONE
C
C  AUTHOR:  W.J. DICKSON, JET
C
C  DATE:    20/10/89
C
C UNIX-IDL PORT:
C    WILLIAM OSBORN, TESSELLA SUPPORT SERVICES PLC.
C
C DATE:    22ND APRIL 1996
C
C VERSION: 1.1 DATE: 22-04-96
C MODIFIED: WILLIAM OSBORN
C     - FIRST VERSION. NO CHANGES TO IBM CODE.
C VERSION: 1.2 DATE: 19-02-03
C MODIFIED: H.P. SUMMERS
C - EXTENDED RANGE
C VERSION: 1.3 DATE: 19-02-03
C MODIFIED: R. MARTIN DATE: 24-03-03
C - Fixed bugs in last version.
C
C-----------------------------------------------------------------------
      CHARACTER           CHR
```

9.22 i4lgrp: Subroutine i4lgrp from library adaslib

```
      FUNCTION I4LGRP(CHR)
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: I4LGRP *******************
C
C  PURPOSE: RETURNS ANGULAR MOMENTUM QUANTUN NUMBER OF ORBITAL
C           GIVEN IN THE EISSNER SINGLE HEXADECIMAL CHARACTER FORM
C
C  CALLING PROGRAMS: GENERAL USE
C
C  SUBROUTINE:
C
C  INPUT :(C*1)  CHR     = HEX. ORBITAL CHARACTER (EG. 'A')
C
C  OUTPUT:(I*4)  I4LGRP  = DECIMAL ORBITAL CHARACTER (E.G. '0')
C
C  AUTHOR:  W.J.DICKSON
C
C  DATE:    20/10/89
C
C  UPDATE:  19/02/03  H.P. SUMMERS - EXTENDED RANGE
C
C
C UNIX-IDL PORT:
C    WILLIAM OSBORN, TESSELLA SUPPORT SERVICES PLC.
C
C DATE:    22ND APRIL 1996
C
C VERSION: 1.1 DATE: 22-04-96
C MODIFIED: WILLIAM OSBORN
C     - FIRST VERSION. NO CHANGES TO IBM CODE.
C
C VERSION: 1.2 DATE: 19/02/03
C MODIFIED: H.P. SUMMERS
C - EXTENDED RANGE
C
C-----------------------------------------------------------------------
      CHARACTER           CHR
```

9.23 i4ndec: Subroutine i4ndec from library adaslib

```
      FUNCTION I4NDEC(IDEC)
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 FUNCTION: I4NDEC  ********************
C
C  PURPOSE: RETURNS N QUANTUM NUMBER OF AN ORBITAL GIVEN IN THE
C           DECIMAL FORM
C
C
C  INPUT :  (I*4)   IDEC   = DECIMAL ORBITAL NUMBER (EG.'10')
C
C  OUTPUT:  (I*4)   I4NDEC = PRINCIPAL QUANTUM NUMBER FOR ORBITAL
C
C           (I*4)   NGRPA()= SET OF N-VALUES FOR DECIMAL ORBITALS
C
C  SUBROUTINES:  NONE
C
C  AUTHOR:  H. P. SUMMERS, UNIVERSITY OF STRATHCLYDE
C           JA8.08
C           TEL. 0141-553-4196
C
C  DATE:    06/06/96
C
C  UPDATE:  24/07/96  PEB - REMOVE SPURIOUS EXTRA 'END' STATEMENT.
C
C  UPDATE:  19/02/03  H.P. SUMMERS - EXTENDED RANGE
C
C
C UNIX-IDL PORT:
C    WILLIAM OSBORN, TESSELLA SUPPORT SERVICES PLC.
C
C DATE:    20TH AUGUST 1996
C
C VERSION: 1.1 DATE: 20-08-96
C MODIFIED: WILLIAM OSBORN
C     - FIRST VERSION
C
C VERSION: 1.2 DATE: 19-02-03
C MODIFIED: H.P. SUMMERS
C - EXTENDED RANGE
C
C-----------------------------------------------------------------------
      INTEGER             IDEC
```

9.24 i4ngrp: Subroutine i4ngrp from library adaslib

```
      FUNCTION I4NGRP(CHR)
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 FUNCTION: I4NGRP  ********************
C
C  PURPOSE: RETURNS N QUANTUM NUMBER GIVEN IN THE EISSNER SINGLE
C           HEXADECIMAL CHARACTER FORM
C
C
C  INPUT :  (C*1)   CHR    = HEX. ORBITAL CHARACTER (EG. 'A')
C
C  OUTPUT:  (I*4)   I4NGRP = PRINCIPAL QUANTUM NUMBER FOR ORBITAL
C
C           (I*4)   I      = GENERAL INDEX
C
C  SUBROUTINES:  NONE
C  AUTHOR:  W.J. DICKSON, JET
C
C  DATE:    20/10/89
C
C  UPDATE:  19/02/03  H.P. SUMMERS - EXTENDED RANGE
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 22-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - PUT UNDER SCCS CONTROL
C
C VERSION: 1.2                          DATE: 22-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - PUT IN DEFAULT OF ZERO FOR RETURN VALUE.
C
C VERSION: 1.3                          DATE: 19-02-03
C MODIFIED: H.P. SUMMERS
C - EXTENDED RANGE
C
C-----------------------------------------------------------------------
      CHARACTER           CHR
```

9.25 i4pgrp: Subroutine i4pgrp from library adaslib

```
      FUNCTION I4PGRP(CHR)
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 FUNCTION: I4PGRP  ********************
C
C  PURPOSE: RETURNS PARITY OF ORBITAL GIVEN THE EISSNER SINGLE
C           HEXADECIMAL CHARACTER FORM
C
C
C  INPUT :  (C*1)   CHR    = HEX. ORBITAL CHARACTER (EG. 'A')
C
C  OUTPUT:  (I*4)   I4PGRP = PARITY FOR ORBITAL
C
C           (I*4)   I      = GENERAL INDEX
C
C  SUBROUTINES:  NONE
C  AUTHOR:  W.J. DICKSON, JET
C
C  DATE:    20/10/89
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 22-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - PUT UNDER SCCS CONTROL
C
C VERSION: 1.2                          DATE: 22-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - PUT IN DEFAULT OF ZERO FOR RETURN VALUE
C
C VERSION: 1.2                          DATE: 19-02-03
C MODIFIED: H.P. SUMMERS
C - EXTENDED RANGE
C
C-----------------------------------------------------------------------
      CHARACTER           CHR
```

9.26 i4schr: Subroutine i4schr from library adaslib

```
       FUNCTION I4SCHR(CHR,IABT)
C
C-----------------------------------------------------------------
C  ************* FORTRAN77 FUNCTION: I4SCHR *********************
C
C PURPOSE: CONVERT FROM CHARACTER REPRESENTATION OF NUMBER OF
C          EQUIVALENT ELECTRONS TO DECIMAL FORM
C
C  FUNCTION:
C  INPUT : (C*1)   CHR     = CHARACTER FORM FOR THE NO. OF
C                            EQUIVALENT ELECTRONS
C  OUTPUT: (I*4)   I4SCHR  = INTEGER FORM FOR THE NO. OF
C                            EQUIVALENT ELECTRONS
C          (I*4)   IABT    = RETURN CODE:
C                               0 => NO ERROR
C                               1 => ERROR (A VALUE 'I4SCHR=0' WILL BE
C                                           RETURNED).
C          (I*4)   I       = GENERAL INTEGER
C
C  SUBROUTINES: NONE
C
C  AUTHOR: H. P. SUMMERS, UNIVERSITY OF STRATHCLYDE
C
C  DATE:  30/10/95
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 22-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - PUT UNDER SCCS CONTROL
C
C VERSION: 1.2                          DATE: 19-02-03
C MODIFIED: H.P. SUMMERS
C - EXTENDED RANGE
C
C-----------------------------------------------------------------
      CHARACTER           CHR
      INTEGER             IABT
```

9.27 i4unit: Subroutine i4unit from library adaslib

```
      FUNCTION I4UNIT( IUNIT )
C-----------------------------------------------------------------------
C
C  ************** FORTRAN77 INTEGER*4 FUNCTION: I4UNIT *****************
C
C  PURPOSE: TO RESET OR RETURN A STORED INTEGER*4 VALUE GREATER THAN OR
C           EQUAL TO ZERO.
C           THIS IS USED WITHIN ADAS TO STORE THE STREAM/UNIT NUMBER
C           FOR THE OUTPUT OF ERROR MESSAGES (TO THE SCREEN).
C
C           BY DEFAULT THE STORED VALUE WILL BE 6, AND WILL BE RETURNED
C           BY THE FUNCTION IF IUNIT ON INPUT < 0.
C
C           TO RESET THE STORED VALUE THEN SET IUNIT TO THE REQUIRED
C           POSITIVE INTEGER (INC. ZERO). THIS VALUE WILL ALSO BE
C           RETURNED BY THE FUNCTION.
C
C                 IUNIT VALUE               RETURNED FUNCTION VALUE
C                 -----------               -----------------------
C                 IUNIT <  0            = CURRENT STORED INTEGER VALUE
C                                           (6 BY DEFAULT).
C                 IUNIT >= 0            = IUNIT , AND RESETS THE STORED
C                                                 VALUE TO IUNIT.
C
C
C  CALLING PROGRAM: GENERAL USE
C
C  SUBROUTINE:
C
C  O     : (I*4)  I4UNIT   = FUNCTION NAME - (SEE ABOVE)
C
C  I     : (I*4)  IUNIT    = FUNCTION ARGUMENT - (SEE ABOVE)
C
C          (I*4)  IDEFLT   = PARAMETER = DEFAULT STORED INTEGER VALUE
C
C          (I*4)  ICURNT   = CURRENT STORED INTEGER VALUE
C
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/37
C          JET EXT. 5023
C
C DATE:    23/04/93
C
C UPDATE:  24/05/93 - PE BRIDEN - ALLOWED 0 TO BE A VALID STORED NUMBER
C
C-----------------------------------------------------------------------
      INTEGER             IUNIT
```

9.28 ingama: Subroutine ingama from library adaslib

```
      function ingama(a,x)

C-----------------------------------------------------------------------
C
C    ****************** fortran77 function: ingama ******************
C
C  purpose: evaluates incomplete gamma function, P(a,x)
C
C  calling program: various
C
C  input : (r*8)  x      = function argument
C  input : (r*8)  a      = function argument
C
C  output: (r*8)  ingama = function name
C
C  author: Paul Bryans, University of Strathclyde
C
C  date:   20/02/04
C
C  update:
C
C-----------------------------------------------------------------------
      REAL*8              A,           X
```

9.29 ingamq: Subroutine ingamq from library adaslib

```
      function ingamq(a,x)

C-----------------------------------------------------------------------
C
C    ****************** fortran77 function: ingamq ******************
C
C  purpose: evaluates incomplete gamma function, 1-P(a,x)
C
C  calling program: various
C
C  input : (r*8)  x      = function argument
C  input : (r*8)  a      = function argument
C
C  output: (r*8)  ingamq = function name
C
C  author: Paul Bryans, University of Strathclyde
C
C  date:   20/02/04
C
C  update:
C
C-----------------------------------------------------------------------
      REAL*8              A,           X
```

9.30 lenstr: Subroutine lenstr from library adaslib

```
      FUNCTION LENSTR(ASTR)
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 FUNCTION: LENSTR ***********************
C
C     PURPOSE : RETURNS THE EFFECTIVE LENGTH OF A GIVEN STRING
C               (IGNORING TRAILING BLANKS)
C
C NOTES: THIS ROUTINE IS NOT YET PROPERLY ANNOTATED
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 18-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - PUT UNDER SCCS CONTROL
C
C VERSION  : 1.2
C DATE     : 10-04-2007
C MODIFIED : Allan Whiteford
C               - Modified documentation as part of automated
C   subroutine documentation preparation.
C-----------------------------------------------------------------------
C
      CHARACTER*(*)       ASTR
```

9.31 linfit: Subroutine linfit from library adaslib

```
       SUBROUTINE LINFIT(X,XSA,Y,YSA,ICT)
C
       IMPLICIT REAL*8(A-H,O-Z)
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: LINFIT *********************
C
C PURPOSE:
C  SUBROUTINE TO PERFORM LINEAR INTERPOLATION
C
C
C  INPUT
C      X     = REQUIRED X-VALUE
C      XSA(I)= X-VALUES
C      YSA(I)= Y-VALUES
C      ICT   = NUMBER OF VALUES
C  OUTPUT
C      Y     = RETURNED Y-VALUE
C
C NOTES: THIS ROUTINE IS NOT YET PROPERLY ANNOTATED
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 06-03-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - PUT UNDER S.C.C.S. CONTROL
C
C-----------------------------------------------------------------------
C
      INTEGER             ICT
      REAL*8              X,           XSA(10),     Y,           YSA(10)
```

9.32 lngama: Subroutine lngama from library adaslib

```
      function lngama(x)

C-----------------------------------------------------------------------
C
C    ****************** fortran77 function: lngama ******************
C
C  purpose: Returns the natural logarithm of the Gamma function of x
C
C  calling program: various
C
C  input : (r*8)  x      = function argument
C
C  output: (r*8)  lngama = function name
C
C  author: Paul Bryans, University of Strathclyde
C
C  date:   26/01/04
C
C  update:
C
C-----------------------------------------------------------------------
      REAL*8              X
```

9.33 matin1: Subroutine matin1 from library adaslib

```
      SUBROUTINE MATIN1(N,A,B)
C-----------------------------------------------------------------------
C  ******************FORTRAN77 SUBROUTINE: MATIN1 *********************
C
C  PURPOSE: MATRIX INVERSION
C
C  CALLING PROGRAM:
C           LSTSQ
C
C  INPUT:
C           N  NUMBER OF DATA POINTS
C
C  I/O      (R*8)  A(,) 'N' BY 'N' MATRIX
C                           INPUT : MATRIX TO BE INVERTED
C                           OUTPUT: INVERTED MATRIX
C           (R*8)  B()  = A.X = B
C                           INPUT : RIGHT HAND SIDE VECTOR 'B'
C                           OUTPUT: SOLUTION VECTOR 'X'
C
C  DATE: 02/07/95 VERSION 1.1
C  WRITTEN: A.LANZAFAME & D.H.BROOKS, UNIV.OF STRATHCLYDE
C           CONVERTED FROM BURGESS BBC BASIC
C
C-----------------------------------------------------------------------
      INTEGER             N
      REAL*8              A(N,N),      B(N)
```

9.34 matinv: Subroutine matinv from library adaslib

```
      SUBROUTINE MATINV(A,N,B,M,DETERM)
      IMPLICIT REAL*8 (A-H,O-Z)
C
C-----------------------------------------------------------------------
C
C  ***************** FORTRAN77 PROGRAM: MATINV ***********************
C
C PURPOSE: MATRIX INVERSION WITH ACCOMPANYING SOLUTION
C          OF LINEAR EQUATIONS
C
C NOTES: THIS ROUTINE IS NOT YET PROPERLY ANNOTATED
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 12-12-95
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST VERSION
C
C VERSION  : 1.2
C DATE     : 10-04-2007
C MODIFIED : Allan Whiteford
C               - Modified documentation as part of automated
C   subroutine documentation preparation.
C
C-----------------------------------------------------------------------
C
      INTEGER             M,           N
      REAL*8              A(30,30),    B(30),       DETERM
```

9.35 nsort: Subroutine nsort from library adaslib

```
      SUBROUTINE NSORT(XA,IA,N)
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: NSORT **********************
C
C PURPOSE:
C SUBROUTINE TO SORT AN ARRAY SO THAT XA IS INCREASING ORDER
C
C
C INPUT
C     XA(I)=X-VALUES
C     IA(I)=I-VALUES
C     N=NUMBER OF VALUES
C OUTPUT
C     XA(I)=SORTED X-VALUES
C     IA(I)=SORTED I-VALUES
C
C NOTES: THIS ROUTINE IS NOT YET PROPERLY ANNOTATED
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 01-02-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - PUT UNDER SCCS CONTROL
C
C-----------------------------------------------------------------------
      INTEGER             IA(*),       N
      REAL                XA(*)
```

9.36 r8ah: Subroutine r8ah from library adaslib

```
      FUNCTION R8AH ( NU , LU , NL , LL )
C
C
C-----------------------------------------------------------------------
C
C  ***************** FORTRAN77 REAL*8 FUNCTION: R8AH   ******************
C
C  PURPOSE: CALCULATES A-VALUES FOR HYDROGEN.
C
C  CALLING PROGRAM: GENERAL USE
C
C  FUNC:   (R*8)  R8AH    = HYDROGEN A-VALUE.
C
C  INPUT:  (I*4)  NU      = UPPER VALUE OF N QUANTUM NUMBER.
C  INPUT:  (I*4)  LU      = UPPER VALUE OF L QUANTUM NUMBER.
C  INPUT:  (I*4)  NL      = LOWER VALUE OF N QUANTUM NUMBER.
C  INPUT:  (I*4)  LL      = LOWER VALUE OF L QUANTUM NUMBER.
C
C  PARAM:  (R*8)  P1      = EQUATION CONSTANT.
C
C          (R*8)  XNU     = REAL VALUE = NU.
C          (R*8)  XLU     = REAL VALUE = LU.
C          (R*8)  XNL     = REAL VALUE = NL.
C          (R*8)  XLL     = REAL VALUE = LL.
C          (R*8)  T1      =
C          (R*8)  DE      =
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          R8RD2B     ADAS      RETURNS HYDRONIC BOUND-BOUND RADIAL
C                               INTEGRAL.
C
C AUTHOR:   JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C           K1/0/81
C           JET EXT. 5183
C
C DATE:     29/09/93
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             LL,          LU,          NL,          NU
```

9.37 r8atab: Subroutine r8atab from library adaslib

```
      FUNCTION R8ATAB ( IZ1 , NU , LU , NL , LL )
C
C
C-----------------------------------------------------------------------
C
C  ***************** FORTRAN77 REAL*8 FUNCTION: R8ATAB ******************
C
C  PURPOSE: CALCULATES HYDRONIC L RESOLVED A-VALUES.
C
C           ON FIRST CALL WITH VALID INPUT PARAMETERS, FUNCTION SETS UP
C           A TABLE OF A-VALUES. ON SUBSEQUENT CALLS VALUES ARE THEN
C           LOOKED UP.
C
C           FUNCTION CHECKS TO SEE IF A-VALUE IS POSSIBLE AND DIPOLE
C           ALLOWED AND RETURNS ZERO IF NOT.
C
C  CALLING PROGRAM: GENERAL USE
C
C  FUNC:    (R*8)  R8ATAB    = HYDROGEN A-VALUE.
C
C  INPUT:   (I*4)  IZ1       = ION CHARGE.
C  INPUT:   (I*4)  NU        = UPPER VALUE OF N QUANTUM NUMBER.
C  INPUT:   (I*4)  LU        = L QUANTUM NUMBER FOR NU.
C  INPUT:   (I*4)  NL        = LOWER VALUE OF N QUANTUM NUMBER.
C  INPUT:   (I*4)  LL        = L QUANTUM NUMBER FOR NL.
C
C  PARAM:   (I*4)  MXN       = MAXIMUM VALUE OF N QUANTUM NUMBER.
C
C           (I*4)  M         = TABLE INDEX.
C           (I*4)  N1        = N QUANTUM NUMBER LOOP INDEX.
C           (I*4)  L1        = N QUANTUM NUMBER LOOP INDEX.
C           (I*4)  N2        = N QUANTUM NUMBER LOOP INDEX.
C           (I*4)  L2        = N QUANTUM NUMBER LOOP INDEX.
C           (I*4)  J0        = TABLE INDEX.
C
C           (R*8)  Z14       = REAL VALUE = IZ1**4
C
C           (L*4)  LFIRST    = .TRUE.  = FIRST TIME FUNCTION CALLED.
C                              .FLASE. = FUNCTION HAS BEEN CALLED BEFORE.
C
C           (I*4)  MA()      = INDEX TABLE FOR 'ATABLE'.
C                              DIMENSION: REFERENCED BY NU QUANTUM NO.
C           (R*8)  ATABLE()  = TABLE OF A-VALUES.
C                              DIMENSION: REFERENCED BY NU, LU, NL, LL
C                                         QUANTUM NUMBERS.
C
C ROUTINES:
C           ROUTINE    SOURCE    BRIEF DESCRIPTION
C           ------------------------------------------------------------
C           I4UNIT     ADAS      RETURNS UNIT NO. FOR OUPUT OF MESSAGES.
C           R8AH       ADAS      RETURNS HYDRONIC A-VALUE.
C
C AUTHOR:    JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C            K1/0/81
C            JET EXT. 5183
C
C DATE:      29/09/93
C
C-----------------------------------------------------------------------
```

```
C
C-----------------------------------------------------------------------
      INTEGER             IZ1,          LL,           LU,           NL
      INTEGER             NU
```

9.38 r8bcon: Subroutine r8bcon from library adaslib

```
      FUNCTION R8BCON( INTYP , OUTTYP , EIN )
C-----------------------------------------------------------------------
C
C  ***************** FORTRAN77 REAL*8 FUNCTION: R8ECON ******************
C
C  PURPOSE: TO CONVERT A BEAM ENERGY INTO SPECIFIED UNITS
C           (DOUBLE PRECISION FUNCTION VERSION OF 'XXBCON')
C
C  CALLING PROGRAM: GENERAL USE
C
C  FUNCTION:
C
C             (R*8)  R8BCON  = FUNCTION NAME -
C                              OUTPUT BEAM ENERGY (STATED UNITS)
C             (I*4)  INTYP   = 1 => 'EIN' UNITS: EV/AMU
C                              = 2 => 'EIN' UNITS: AT. UNITS
C                              = 3 => 'EIN' UNITS: CM S-1.
C             (I*4)  OUTTYP  = 1 => 'R8BCON' UNITS: EV/AMU
C                              = 2 => 'R8BCON' UNITS: AT. UNITS
C                              = 3 => 'R8BCON' UNITS: CM S-1.
C             (R*8)  EIN     = INPUT BEAM ENERGY (STATED UNITS)
C
C             (R*8)  EVUATU  = EV/AMU  TO AT. UNITS.
C             (R*8)  EVUCMS  = EV/AMU TO CM S-1.
C
C             (R*8)  BCONV() = BEAM ENERGY CONVERSION PARAMETERS
C
C ROUTINES:  NONE
C
C NOTE:
C            BEAM ENERGY CONVERSION PARAMETERS:
C
C            INTYP = 1 ; BCONV(1) =>             EV/AMU  -> OUTPUT UNITS
C            INTYP = 2 ; BCONV(2) =>          AT. UNITS  -> OUTPUT UNITS
C            INTYP = 3 ; BCONV(3) =>             CM S-1  -> OUTPUT UNITS
C
C AUTHOR:  H. P. SUMMERS, UNIVERSITY OF STRATHCLYDE
C          JA8.08
C          TEL. 0141-553-4196
C
C DATE:    21/04/95
C
C UPDATE:  15/05/95 Tim Hammond UNIX PORT
C Put under SCCS control
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             INTYP,       OUTTYP
      REAL*8              EIN
```

9.39 r8const: Subroutine r8const from library adaslib

```
      FUNCTION R8CONST( key )
C-----------------------------------------------------------------------
C
C  ***************** FORTRAN77 REAL*8 FUNCTION: R8const *****************
C
C  PURPOSE: Returns the fundamental constant corresponding to 'KEY'.
C
C  CALLING PROGRAM: GENERAL USE
C
C  FUNCTION:
C
C  INPUT:  (C*)   KEY   =  Constant selector key
C
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4UNIT     ADAS      RETURNS UNIT NO. FOR OUTPUT OF MESSAGES.
C
C
C NOTES:   i. For Rydberg constant XXRAMS (or XXDAMS) is called to
C             determine whether the infinite or mass adjusted rydberg
C             constant is returned.
C
C         ii. Data is taken from Cohen & Taylor, Rev. Mod. Phys. 59,
C             p1121 (1987). As recommended in Atomic Data and Nuclear
C             Data Tables.
C
C        iii. Units are CGS.
C
C
C AUTHOR:  Martin O'Mullane
C
C DATE:    01/04/99
C
C UPDATE:
C
C VERSION:  1.2
C MODIFIED: MARTIN OMULLANE
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      CHARACTER*(*)       KEY
```

9.40 r8dcon: Subroutine r8dcon from library adaslib

```
      FUNCTION R8DCON( INTYP, OUTTYP, IZ1, DIN )
C-----------------------------------------------------------------------
C
C  ***************** FORTRAN77 REAL*8 FUNCTION: R8DCON ******************
C
C  PURPOSE: TO CONVERT A DENSITY INTO SPECIFIED UNITS
C
C           (DOUBLE PRECISION FUNCTION VERSION OF 'XXTCON')
C
C  CALLING PROGRAM: GENERAL USE
C
C  FUNCTION:
C
C             (R*8)  R8DCON  = FUNCTION NAME -
C                              OUTPUT DENSITY (STATED UNITS)
C             (I*4)  INTYP   = 1 => 'DIN' UNITS: CM-3
C                            = 2 => 'DIN' UNITS: REDUCED
C             (I*4)  OUTTYP  = 1 => 'R8DCON' UNITS: CM-3
C                            = 2 => 'R8DCON' UNITS: REDUCED
C             (I*4)  IZ1     = RECOMBINING ION CHARGE (= Z+1).
C             (R*8)  DIN     = INPUT DENSITY (STATED UNITS)
C
C             (R*8)  Z1P7    = 'IZ1'**7
C             (R*8)  DCONV() = DENSITY CONVERSION PARAMETERS
C
C ROUTINES:  NONE
C
C NOTE:
C            DENSITY CONVERSION PARAMETERS:
C
C            INTYP = 1 ; DCONV(1) =>               CM-3  -> OUTPUT UNITS
C            INTYP = 2 ; DCONV(2) =>            REDUCED  -> OUTPUT UNITS
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 4569
C
C DATE:    04/01/91
C
C UPDATE:  07/08/91 - PE BRIDEN: CHANGED 'DBLE(IZ1**7)' TO 'DBLE(IZ1)**7
C                                TO AVOID INTEGER OVERFLOW IF IZ1>21.
C
C VERSION  : 1.2
C DATE     : 10-04-2007
C MODIFIED : Allan Whiteford
C               - Modified documentation as part of automated
C   subroutine documentation preparation.
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             INTYP,       IZ1,         OUTTYP
      REAL*8              DIN
```

9.41 r8ecip: Subroutine r8ecip from library adaslib

```
      FUNCTION R8ECIP( IZC , XI , ZETA , TE )
C
C
C-----------------------------------------------------------------------
C
C  ***************** FORTRAN77 REAL*8 FUNCTION: R8ECIP ******************
C
C  PURPOSE:  CALUCLATES THE SHELL CONTRIBUTION TO THE IONISATION RATE
C            COEFFICIENT IN THE ECIP APPROXIMATION OF BURGESS.
C
C            CF. SUMMERS (1974) APPLETON LABORATORY REPORT IM367.
C
C  CALLING PROGRAM: GENERAL USE
C
C  FUNC:   (R*8)  R8ECIP  = IONISATION RATE COEFFICIENT.
C                           UNITS: CM3 SEC-1
C
C  INPUT : (I*4)  IZC     = TARGET ION CHARGE NUMBER.
C  INPUT : (R*8)  XI      = EFFECTIVE IONISATION POTENTIAL FOR SHELL.
C                           UNITS: RYD
C  INPUT : (R*8)  ZETA    = EFFECTIVE NUMBER OF EQUIVALENT ELECTRONS FOR
C                           SHELL.
C  INPUT : (R*8)  TE      = ELECTRON TEMPERATURE.
C                           UNITS: K
C
C  PARAM : (I*4) MXT      = 5.
C  PARAM : (R*8) P1       =
C
C          (I*4)  J       = LOOP INDEX
C
C          (R*8)  Z       = REAL VALUE = IZC+1.
C          (R*8)  ATE     =
C          (R*8)  EN      =
C          (R*8)  Y       =
C          (R*8)  AI      =
C          (R*8)  B       =
C          (R*8)  B1      =
C          (R*8)  C       =
C          (R*8)  R       =
C          (R*8)  D       =
C          (R*8)  F       =
C          (R*8)  C1      =
C          (R*8)  C2      =
C          (R*8)  C4      =
C
C          (R*8)  X()     =
C          (R*8)  W()     =
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          R8YIP      ADAS
C
C AUTHOR:  JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 5183
C
C DATE:    03/11/93
C
```

```
C UNIX-IDL PORT:
C
C AUTHOR:  WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C
C DATE:    22ND MAY 1996
C
C VERSION: 1.1                          DATE: 22-05-96
C MODIFIED: WILLIAM OSBORN
C               - FIRST VERSION. IBM VERSION NOT CHANGED
C
C VERSION: 1.2                          DATE: 10-04-2007
C MODIFIED : Allan Whiteford
C               - Modified documentation as part of automated
C   subroutine documentation preparation.
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IZC
      REAL*8              TE,          XI,          ZETA
```

9.42 r8econ: Subroutine r8econ from library adaslib

```
      FUNCTION R8ECON( INTYP, OUTTYP, EIN )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: R8ECON *********************
C
C  PURPOSE: TO CONVERT A VELOCITY/ENERGY INTO A SPECIFIED FORM
C
C           (DOUBLE PRECISION FUNCTION VERSION OF 'XXECON')
C
C  CALLING PROGRAM: GENERAL USE
C
C  FUNCTION:
C
C             (R*8)  R8ECON  = FUNCTION NAME -
C                              OUTPUT VELOCITY/ENERGY (STATED UNITS)
C             (I*4)  INTYP   = 1 => 'EIN' UNITS: AT. UNITS (VEL.)
C                            = 2 => 'EIN' UNITS: CM/SEC    (VEL.)
C                            = 3 => 'EIN' UNITS: EV/AMU  (ENERGY)
C             (I*4)  OUTTYP  = 1 => 'R8ECON' UNITS: AT.UNITS (VEL.)
C                            = 2 => 'R8ECON' UNITS: CM/SEC   (VEL.)
C                            = 3 => 'R8ECON' UNITS: EV/AMU (ENERGY)
C             (R*8)  EIN     = INPUT  VELOCITY/ENERGY (STATED UNITS)
C
C             (R*8)  AMU2KG  = PARAMETER: AMU TO KG CONVERSION FACTOR
C             (R*8)  EV2J    = PARAMETER: EV TO JOULES CONVERSION FACTOR
C             (R*8)  M2CM    = PARAMETER: METRES TO CM CONVERSION FACTOR
C             (R*8)  VELE0H  = PARAMETER: ORBITAL VELOCITY  (CM/SEC)  OF
C                              AN ELECTRON IN THE SMALLEST  ORBIT  OF  A
C                              HYDROGEN ATOM (BOHR) = 2.1877D+8 CM/SEC
C             (R*8)  AT2VEL  = AT.UNITS (VEL) TO CM/SEC (VEL) CONVERSION
C             (R*8)  VEL2AT  = CM/SEC (VEL) TO AT.UNITS (VEL) CONVERSION
C             (R*8)  VEL2EN  = CM/SEC (VEL) TO EV/AMU (ENGY.) CONVERSION
C             (R*8)  EN2VEL  = EV/AMU (ENGY.) TO CM/SEC (VEL) CONVERSION
C             (R*8)  AT2EN   = AT.UNITS (VEL) TO EV/AMU (ENG) CONVERSION
C             (R*8)  EN2AT   = EV/AMU (ENG) TO AT.UNITS (VEL) CONVERSION
C
C             (R*8)  ECONV() = ENERGY/VELOCITY CONVERSION PARAMETERS
C
C ROUTINES:  NONE
C
C NOTE:
C            ENERGY/VELOCITY CONVERSION PARAMETERS:
C
C            INTYP = 1 ; ECONV(1) =>  VELOCITY: AT.UNITS -> OUTPUT FORM
C            INTYP = 2 ; ECONV(2) =>  VELOCITY: CM/SEC   -> OUTPUT FORM
C            INTYP = 3 ; ECONV(3) =>  ENERGY  : EV/AMU   -> OUTPUT FORM
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 4569
C
C DATE:    05/02/91
C
C VERSION: 1.2 DATE: 01-05-96
C MODIFIED: WILLIIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C     - CORRECTED AT2EN. WAS AT2VEL*VEL2EN
C
C VERSION: 1.2                          DATE: 10-04-2007
C MODIFIED : Allan Whiteford
```

```
C               - Modified documentation as part of automated
C   subroutine documentation preparation.
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             INTYP,       OUTTYP
      REAL*8              EIN
```

9.43 r8erfc: Subroutine r8erfc from library adaslib

```
       function r8erfc( x )
c-----------------------------------------------------------------------
c
c  ****************** fortran77 function: r8erfc *********************
c
c  purpose:  calculates the error function erfc(x)
c
c  calling program: various
c
c
c
c  input : (i*4)  x      = independent variable.
c  output: (r*8)  r8erfc = erfc(x).
c
c  routines:
c          none
c
c  author:  Hugh Summers, University of Strathclyde
c           JA7.08
c           tel. 0141-548-4196
c
c  date:    15/01/02
c
c  update:
c
c-----------------------------------------------------------------------
      REAL*8              X
```

9.44 r8expe: Subroutine r8expe from library adaslib

```
      REAL*8 FUNCTION R8EXPE( VALUE , EXPON , CODE )
C-----------------------------------------------------------------------
C
C  ***************** FORTRAN77 REAL*8 FUNCTION: R8EXPE ******************
C
C  PURPOSE: TO PRODUCE A NUMBER FROM VALUE AND EXPONENT PARTS PROTECTED
C           AGAINST UNDERFLOW AND OVERFLOW. (NUMBER=VALUE*EXP(EXPON))
C
C  INPUT : (R*8)  VALUE   = VALUE PART OF COMPOSITE NUMBER
C  INPUT : (R*8)  EXPON   = EXPONENT OF COMPOSITE NUMBER
C  INPUT : (C*(*))CODE    = 'ZERO' => ZERO VALUE DELIVERED IF UNDERFLOW
C                           'MIN'  => MINIMUM VALUE DELIVERED
C  OUTPUT: (R*8)  R8EXPE  = EVALUATED NUMBER
C
C  ROUTINES:
C           NONE
C
C  AUTHOR:   H. P. SUMMERS, UNIVERSITY OF STRATHCLYDE
C            TEL. 0141-548-4196
C            JET EXT. 5057
C
C  DATE:     16/09/99
C
C  UPDATE:
C
C VERSION: 1.1
C
C-----------------------------------------------------------------------
      CHARACTER*(*)       CODE
      REAL*8              EXPON,       VALUE
```

9.45 r8f21: Subroutine r8f21 from library adaslib

```
C
      function r8f21(a,b,c,d,eps)
C-----------------------------------------------------------------------
C
C  ****************** fortran77 function: r8f21.for ********************
C
C  Purpose:  Evaluates series expansion of hypergeometric function
C            F(a,b;c;d).
C
C  Function:
C
C  Input : (r*8)  a        = 1st numerator parameter a
C  Input : (r*8)  b        = 2nd numerator parameter b
C  Input : (r*8)  d        = 3rd denominator parameter c
C  Input : (r*8)  d        = independent variable (equiv x)
C  Input : (r*8)  eps      = accuracy parameter
C
C  Output: (r*8)  r8f21    = F(a,b;c;d)
C
C  Routines:
C          Routine    Source    Brief description
C          ------------------------------------------------------------
C          i4unit     ADAS     Specifies unit for warning message ouput
C
C  Author:  H. P. summers, University of Strathclyde
C           ja7.08
C           tel. 0141-548-4196
C
C  Date:   24/02/03
C
C  Update: HP Summers    24/05/04
C          restructured and added standard warning
C
C  Update: AD Whiteford  16/03/05
C          renamed to r8f21
C
C  Update: HP Summers    17/04/07
C          Corrected description comments
C
C-----------------------------------------------------------------------
      REAL*8              A,           B,           C,           D
      REAL*8              EPS
```

9.46 r8fbch: Subroutine r8fbch from library adaslib

```
      FUNCTION R8FBCH( IZ , XI , ZETA , TE )
C-----------------------------------------------------------------------
C
C  ***************** FORTRAN77 REAL*8 FUNCTION: R8FBCH ******************
C
C  PURPOSE: EVALUATES A SHELL CONTRIBUTION TO THE IONISATION RATE COEFF-
C           ICIENT IN THE BURGESS-CHIDICHIMO APPROXIMATION.
C
C  REFERENCE: MNRAS.(1983)203,1269.
C
C  CALLING PROGRAM: GENERAL USE
C
C  FUNCTION:
C          (R*8)  R8FBCH  = FUNCTION NAME
C          (I*4)  IZ      = TARGET ION CHARGE NUMBER
C                           (RECOMBINED ION CHARGE).
C          (R*8)  XI      = EFFECTIVE IONISATION POTENTIAL FOR SHELL
C                           (UNITS: RYDBERGS)
C                           (LEVEL ENERGY RELATIVE TO IONISATION POT.)
C          (R*8)  ZETA    = EFFECTIVE NUMBER OF EQUIVALENT ELECTRONS
C                           IN SHELL
C          (R*8)  TE      = ELECTRON TEMPERATURE (IN KELVIN)
C
C          (R*8)  C       = PARAMETER = EQUATION CONSTANT = 2.3
C          (R*8)  DXIPOW  = PARAMETER = EQUATION CONSTANT = 1.5
C          (R*8)  TK2ATE  = PARAMETER = EQUATION CONSTANT = 1.5789D+05
C          (R*8)  R2GAM   = PARAMETER = EQUATION CONSTANT = 2.17161D-08
C          (R*8)  D150    = PARAMETER = 150
C          (R*8)  CR2GAM  = PARAMETER = EQUATION CONSTANT = 'C'*'R2GAM'
C
C          (R*8)  BETA    = EQUATION CONSTANT (SEE NOTE BELOW)
C          (R*8)  Y       =
C          (R*8)  T1      =
C          (R*8)  W       =
C          (R*8)  P       = TEMPORARY PARAMETER STORAGE
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          R8FEEI     ADAS      FUNCTION:
C
C NOTE:
C                              (100*IZ) + 91
C                     SQRT( ------------- ) - 5
C                              (4*IZ) + 3
C           BETA =  -------------------------
C                                 4
C
C
C AUTHOR:   PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C           K1/0/81
C           JET EXT. 4569
C
C DATE:     04/09/90
C
C VERSION  : 1.2
C DATE     : 20-12-2001
C MODIFIED : Martin O'Mullane
C               - Removed mainframe listing information beyond column 72.
```

```
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      INTEGER             IZ
      REAL*8              TE,          XI,          ZETA
```

9.47 r8fctn: Subroutine r8fctn from library adaslib

```
      FUNCTION R8FCTN( STR , IABT )
C-----------------------------------------------------------------------
C
C  **************** FORTRAN77 REAL*8 FUNCTION: R8FCTN ******************
C
C  PURPOSE : TO CONVERT A FLOATING POINT NUMBER STORED IN A STRING
C            INTO A REAL*8 VARIABLE.
C
C  CALLING PROGRAM: GENERAL USE
C
C  FUNCTION:
C
C          (R*8)   R8FCTN  = FUNCTION NAME
C          (C*(*)) STR     = STRING CONTAINING SINGLE FLOATING POINT NO.
C          (I*4)   IABT    = RETURN CODE:
C                                0 => NO ERROR
C                                2 => ERROR (A VALUE 'R8FCTN=0.0' WILL BE
C                                           RETURNED).
C                                9 => OVERFLOW ERROR (EXPONENT  > IUOFLW)
C                                     (A VALUE 'R8FCTN=0.0' RETURNED)
C                               10 => UNDERFLOW ERROR (EXPONENT <-IUOFLW)
C                                     (A VALUE 'R8FCTN=0.0' RETURNED)
C
C          (I*4)   IUOFLW   = PARAMETER = MODULUS OF MAXIMUM ALLOWED
C                                         EXPONENT = 60
C
C          (C*1)   CH0      = PARAMETER = '0'
C          (C*1)   CH9      = PARAMETER = '9'
C          (C*1)   BLANK    = PARAMETER = ' '
C          (C*1)   CPLUS    = PARAMETER = '+'
C          (C*1)   CMINUS   = PARAMETER = '-'
C          (C*1)   CPNT     = PARAMETER = '.'
C          (C*1)   CHE      = PARAMETER = 'E'
C          (C*1)   CHD      = PARAMETER = 'D'
C          (C*1)   CLE      = PARAMETER = 'e'
C          (C*1)   CLD      = PARAMETER = 'd'
C
C          (I*4)   ILEN     = LENGTH OF 'STR' STRING IN BYTES
C          (I*4)   M1       = STARTING BYTE IN 'STR' OF NUMBER
C                             INCLUDING SIGN
C          (I*4)   M2       = LAST BYTE IN 'STR' OF NUMBER
C          (I*4)   IE       = STARTING BYTE OF EXPONENT IN  'STR'
C                             IGNORING ANY SIGN PRESENT.
C          (I*4)   MS       = 0 => MANTISSA HAS NO SIGN
C                             1 => MANTISSA HAS A SIGN
C          (I*4)   IS       = 0 => EXPONENT HAS NO SIGN
C                             1 => EXPONENT HAS A SIGN
C          (I*4)   IPOW     = EXPONENT
C          (I*4)   ICH0     = ICHAR('0')
C          (I*4)   ICH9     = ICHAR('9')
C          (I*4)   ISTR     = ICHAR(CURRENT BYTE POSITION IN 'STR')
C          (I*4)   I        = GENERAL USE
C
C          (L*4)   LMANT    = .TRUE.  => MANTISSA BEING ANALYSED
C                             .FALSE. => EXPONENT BEING ANALYSED
C          (L*4)   LPOINT   = .TRUE.  => DECIMAL POINT FOUND IN MANTISSA
C                             .FALSE. => NO DECIMAL POINT FOUND IN MANT.
C          (L*4)   LFOUND   = .TRUE.  => ALL OF THE INPUT NUMBER BYTES
C                                        HAVE BEEN ASSESSED.
```

```
C                              .FALSE. => INPUT NUMBER BYTES STILL BEING
C                                         ASSESSED.
C
C NOTE:     AN ERROR WILL OCCUR (IABT=2) IF THERE IS MORE THAN ONE
C           NUMBER OCCURING IN THE STRING 'STR()'
C
C
C AUTHOR:   PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C           K1/0/81
C           JET EXT. 4569
C
C DATE:     26/10/90
C
C VERSION  : 1.2
C DATE     : 20-12-2001
C MODIFIED : Martin O'Mullane
C               - Removed mainframe listing information beyond column 72.
C
C VERSION  : 1.3
C DATE     : 03-12-2003
C MODIFIED : Hugh Summers
C               - Allowed lower case 'e' or 'd' in the real number spec.
C
C VERSION  : 1.4
C DATE     : 10-04-2007
C MODIFIED : Allan Whiteford
C               - Modified documentation as part of automated
C   subroutine documentation preparation.
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      CHARACTER*(*)       STR
      INTEGER             IABT
```

9.48 r8fdip: Subroutine r8fdip from library adaslib

```
C
      REAL*8 FUNCTION R8FDIP(E1,L1,E2,L2)
      IMPLICIT REAL*8(A-H,O-Z)
C-----------------------------------------------------------------------
C
C  ***************** FORTRAN77 FUNCTION: R8FDIP **********************
C
C PURPOSE: CALCULATES THE DIPOLE INTEGRAL I(KAPPA1,L1,KAPPA2,L2,1)
C
C NOTE: CREATED BY  ALAN BURGESS AS DEFINED IN PHIL. TRANS. ROY. SOC.
C       A226,255,1970, WHERE E1=KAPPA1**2 AND E2=KAPPA2**2. APPLIES TO
C       POSITIVE ELECTRON ENERGIES, THAT IS THE FREE-FREE CASE.
C       IT IS SUITABLE FOR USE IN EQUATIONS (8),(9),(10) OR (11) OF
C       J. PHYS. B. 7,L364,1974.
C
C CALLING PROGRAMS: GENERAL
C
C INPUT:  (R*8)  E1      = KAPPA1**2 WHERE KAPPA1 IS SCALED INITIAL
C                          ELECTRON WAVE NUMBER
C INPUT:  (I*4)  L1      = ORBITAL ANGULAR OMENTUM OF INITIAL ELECTRON
C INPUT:  (R*8)  E2      = KAPPA2**2 WHERE KAPPA2 IS SCALED INITIAL
C                          ELECTRON WAVE NUMBER
C INPUT:  (I*4)  L2      = ORBITAL ANGULAR OMENTUM OF FINAL ELECTRON
C
C OUTPUT: (R*8)  R8FDIP  = I(KAPPA1,L1,KAPPA2,L2,1)
C
C ROUTINES:
C          ROUTINE    SOURCE   BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          R8FDIP1    ADAS     SMALL ENRGY CHGE.APPROX. TO DIPOLE INTEG.
C          R8FDIP2    ADAS     SMALL ENRGY CHGE.APPROX. TO DIPOLE INTEG.
C          R8FMON1    ADAS     EVALUATES MONOPOLE INTEGRAL
C          ARGAM      ADAS     CALCULATES ARGGAMMA(L+1+I*A)
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 17-04-07
C MODIFIED: HUGH SUMMERS
C               - FIRST FULLY COMMENTED RELEASE
C
C-----------------------------------------------------------------------
      IF(E1+E2-1.0D-40) 11,11,12
   11 R8FDIP=0.0D0
      RETURN
   12 IF(E1-E2) 1,1,2
    1 EMIN=E1
      EMAX=E2
      GO TO 3
    2 EMIN=E2
      EMAX=E1
    3 T=EMIN/EMAX
      IF(T-0.02944D0) 4,4,5
    4 R8FDIP=R8FDIP1(E1,L1,E2,L2)
      GO TO 9
    5 IF(T-0.16667D0) 7,6,6
    6 R8FDIP=R8FDIP2(E1,L1,E2,L2)
      GO TO 9
    7 R8FDIP=R8FDIP1(E1,L1,E2,L2)
       IF(R8FDIP*R8FDIP-1.0D-40) 6,6,8
```

```
  8 RETURN
  9  IF(R8FDIP*R8FDIP-1.0D-40) 10,10,8
 10 WRITE(6,100)
    RETURN
100 FORMAT(17H   R8FDIP FAILURE)
    END
    INTEGER             L1,          L2
    REAL*8              E1,          E2
```

9.49 r8fdip0: Subroutine r8fdip0 from library adaslib

```
      function r8fdip0(e1,l1,e2,l2,eps)
C-----------------------------------------------------------------------
C
C  ****************** fortran77 program: r8fdip0.for ******************
C
C  purpose: calculates the function i0(k1,l1,k2,l2,1) defined in
C           Phil. Trans. Roy. Soc. a266,255,1970, where
C           e1=k1*k1, e2=k2*k2, and the relative accuracy is
C           approximately eps.
C
C           It is suitable for use in equations (13) etc. of
C           J.Phys.B. 7,l364,1974
C           (original by A. Burgess, DAMTP, University of Cambridge)
C
C
C  subroutine:
C
C  input : (r*8)  e1       = initial electron energy (Ryd)
C  input : (r*8)  l1       = inital orbital angular momentum
C  input : (r*8)  e2       = final electron energy (Ryd)
C  input : (r*8)  l2       = final orbital angular momentum
C  input : (r*8)  eps      = accuracy setting
C
C  output: (r*8)  r8fdip0  = dipole matrix element for neutral atom
C
C
C  Routines:
C           routine    source    brief description
C           ------------------------------------------------------------
C           f21        adas      special quadrature for Burgess codes
C           i4unit     adas      fetch unit number for output of messages
C
C
C  Author:  H. P. Summers, University of Strathclyde
C           ja7.08
C           tel. 0141-548-4196
C
C  Date:   24/02/03
C
C  Update: HP Summers    24/05/04
C           restructure and addded standard warning
C  Update: AD Whiteford  16/03/05
C           renamed to r8fdip
C           Now calls r8f21 instead of just f21, this routine was renamed
C  Update: AD Whiteford  17/05/07
C           Updated comments as part of subroutine documentation
C           procedure.
C
C-----------------------------------------------------------------------
      INTEGER             L1,          L2
      REAL*8              E1,          E2,          EPS
```

9.50 r8fdip1: Subroutine r8fdip1 from library adaslib

```
C
       REAL*8 FUNCTION R8FDIP1(E1,L1,E2,L2)
       IMPLICIT REAL*8(A-H,O-Z)
C-----------------------------------------------------------------------
C
C  ***************** FORTRAN77 FUNCTION: R8FDIP1 *********************
C
C PURPOSE: CALCULATES THE DIPOLE INTEGRAL I(KAPPA1,L1,KAPPA2,L2,1) FOR
C          WHERE MIN(E1,E2)/EMAX(E1,E2) < 0.03
C
C NOTE: CREATED BY  ALAN BURGESS FOR USE IN THE DIPOLE INTEGRAL
C       I(KAPPA1,L1,KAPPA2,L2,1) EVALUATION AS DEFINED IN PHIL.
C       TRANS. ROY. SOC. A226,255,1970, WHERE E1=KAPPA1**2 AND
C       E2=KAPPA2**2. APPLIES TO POSITIVE ELECTRON ENERGIES, .
c       THAT IS THE FREE-FREE CASE.
C
C CALLING PROGRAMS: R8FDIP
C
C INPUT:  (R*8)  E1      = KAPPA1**2 WHERE KAPPA1 IS SCALED INITIAL
C                          ELECTRON WAVE NUMBER
C INPUT:  (I*4)  L1      = ORBITAL ANGULAR OMENTUM OF INITIAL ELECTRON
C INPUT:  (R*8)  E2      = KAPPA2**2 WHERE KAPPA2 IS SCALED INITIAL
C                          ELECTRON WAVE NUMBER
C INPUT:  (I*4)  L2      = ORBITAL ANGULAR OMENTUM OF FINAL ELECTRON
C
C OUTPUT: (R*8)  R8FDIP1 = I(KAPPA1,L1,KAPPA2,L2,1)
C
C ROUTINES:
C          ROUTINE    SOURCE   BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          R8FMON1    ADAS     EVALUATES MONOPOLE INTEGRAL
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 17-04-07
C MODIFIED: HUGH SUMMERS
C               - FIRST FULLY COMMENTED RELEASE
C
C-----------------------------------------------------------------------
       IF(L1-L2)1,2,3
    1  L=L1
       A1=E1
       A2=E2
       GO TO 4
    2  R8FDIP1=0.0D0
       RETURN
    3  L=L2
       A1=E2
       A2=E1
    4  LP=L+1
       ELP=LP
       B1=DSQRT(1.0D0+ELP*ELP*A2)*R8FMON1(E1,E2,L)
       B2=DSQRT(1.0D0+ELP*ELP*A1)*R8FMON1(E1,E2,LP)
       IF(B1*B2-1.0D-40)5,5,6
    5  R8FDIP1=0.0D0
       RETURN
    6  R8FDIP1=(B1-B2)/ELP
       RETURN
      END
```

```
INTEGER             L1,          L2
REAL*8              E1,          E2
```

9.51 r8fdip2: Subroutine r8fdip2 from library adaslib

```
C
       REAL*8 FUNCTION R8FDIP2(E1,L1,E2,L2)
       IMPLICIT REAL*8(A-H,O-Z)
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 FUNCTION: R8FDIP2 **********************
C
C PURPOSE: CALCULATES THE DIPOLE INTEGRAL I(KAPPA1,L1,KAPPA2,L2,1) FOR
C          WHERE MIN(E1,E2)/EMAX(E1,E2) > 0.03
C
C NOTE: CREATED BY  ALAN BURGESS FOR USE IN THE DIPOLE INTEGRAL
C       I(KAPPA1,L1,KAPPA2,L2,1) EVALUATION AS DEFINED IN PHIL.
C       TRANS. ROY. SOC. A226,255,1970, WHERE E1=KAPPA1**2 AND
C       E2=KAPPA2**2. APPLIES TO POSITIVE ELECTRON ENERGIES, .
c       THAT IS THE FREE-FREE CASE.
C
C CALLING PROGRAMS: R8FDIP
C
C INPUT:  (R*8)  E1      = KAPPA1**2 WHERE KAPPA1 IS SCALED INITIAL
C                          ELECTRON WAVE NUMBER
C INPUT:  (I*4)  L1      = ORBITAL ANGULAR OMENTUM OF INITIAL ELECTRON
C INPUT:  (R*8)  E2      = KAPPA2**2 WHERE KAPPA2 IS SCALED INITIAL
C                          ELECTRON WAVE NUMBER
C INPUT:  (I*4)  L2      = ORBITAL ANGULAR OMENTUM OF FINAL ELECTRON
C
C OUTPUT: (R*8)  R8FDIP2 = I(KAPPA1,L1,KAPPA2,L2,1)
C
C ROUTINES:
C          ROUTINE    SOURCE   BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          ARGAM      ADAS     CALCULATES ARGGAMMA(L+1+I*A)
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 17-04-07
C MODIFIED: HUGH SUMMERS
C               - FIRST FULLY COMMENTED RELEASE
C
C-----------------------------------------------------------------------
       WMAX=200.0D0
       ETA1=1.0D0/DSQRT(E1)
       ETA2=1.0D0/DSQRT(E2)
       W1=ETA2-ETA1
       PI=3.141592653589793D0
       A=DABS(W1)
       B=PI*A
       IF(B-0.01D0)1,1,2
    1  C=3.0D0/(3.0D0-B*(3.0D0-B*(2.0D0-B)))
       C=DSQRT(C)
       GO TO 5
    2  IF(B-14.0D0)4,3,3
    3  C=DSQRT(B+B)
       GO TO 5
    4  B=B+B
       C1=1.0D0-DEXP(-B)
       C=DSQRT(B/C1)
    5  C=0.5D0*C/DSQRT(ETA1*ETA2)
       C2=ETA1+ETA2
       C1=4.0D0*ETA1*ETA2/(C2*C2)
```

```
      L=L1
      IF(L2-L1)6,6,7
    6 L=L2
      T1=ETA1
      ETA1=ETA2
      ETA2=T1
      W1=-W1
    7 C=C*C1**(L+1)
      U0=L+1
      U1=ETA1
      V0=U0
      V1=-ETA2
      W0=1.0D0
      X0=W1/(C2*C2)
      Y2=-ETA2-ETA2
      Y0=-U0*W1+Y2
      Y1=ETA2*W1
      T1=X0/(1.0D0+W1*W1)
      Z0=U0*T1
      Z1=U1*T1
      T=Z0-Z1*W1
      Z1=Z0*W1+Z1
      Z0=T
      Q0=-1.0D0+Z0*Y0-Z1*Y1
      Q1=Z0*Y1+Z1*Y0
      X=W1*X0
    8 U0=U0+1.0D0
      V0=V0+1.0D0
      W0=W0+1.0D0
      IF(W0-WMAX)21,21,20
   20 R8FDIP2=0.0D0
      RETURN
   21 CONTINUE
      Y0=Y0+Y2
      T=Z0*U0-Z1*U1
      Z1=Z0*U1+Z1*U0
      Z0=T
      T=Z0*V0-Z1*V1
      Z1=Z0*V1+Z1*V0
      Z0=T
      T=Z0*W0-Z1*W1
      Z1=Z0*W1+Z1*W0
      Z0=T
      X0=X/(W0*(W0*W0+W1*W1))
      Z0=Z0*X0
      Z1=Z1*X0
      T0=Z0*Y0-Z1*Y1
      T1=Z0*Y1+Z1*Y0
      Q0=Q0+T0
      Q1=Q1+T1
      T1=T0*T0+T1*T1
      T0=Q0*Q0+Q1*Q1
      IF(T0-1.0D24*T1)8,8,9
    9 J1=0
      J2=L+1
      P=ARGAM(J1,W1)+ARGAM(L,ETA1)-ARGAM(J2,ETA2)
      IW0=W0
      IF(A-1.0D-40)11,11,10
   10 P=P+W1*DLOG(C2/A)
   11 P0=DCOS(P)
      P1=DSIN(P)
```

```
 T=P0*Q0-P1*Q1
 Q1=P0*Q1+P1*Q0
 Q0=T
 R8FDIP2=C*Q1
 RETURN
END
INTEGER             L1,          L2
REAL*8              E1,          E2
```

9.52 r8feei: Subroutine r8feei from library adaslib

```
      FUNCTION R8FEEI ( X )
C-----------------------------------------------------------------------
C
C  **************** FORTRAN77 REAL*8 FUNCTION: R8FEEI ******************
C
C  PURPOSE: EVALUATES EXP(X)E1(X) WHERE E1 IS THE 1ST EXPONENTIAL
C           INTEGRAL
C
C  NOTE:    EXACT DUPLICATE IN OPERATION TO EEI.
C
C  REFERENCE:
C
C  CALLING PROGRAM: GENERAL USE
C
C  FUNCTION:
C
C  INPUT:  (R*8)  X       = INPUT PARAMETER
C
C  OUTPUT: (R*8)  R8FEEI  = FUNCTION NAME
C
C          (R*8)  DHALF   = PARAMETER = 0.5
C          (R*8)  CX0     = PARAMETER = EQUATION CONSTANT
C          (R*8)  CX1     = PARAMETER = EQUATION CONSTANT
C          (R*8)  CX2     = PARAMETER = EQUATION CONSTANT
C          (R*8)  CX3     = PARAMETER = EQUATION CONSTANT
C          (R*8)  CX4     = PARAMETER = EQUATION CONSTANT
C          (R*8)  CX5     = PARAMETER = EQUATION CONSTANT
C          (R*8)  CY0     = PARAMETER = EQUATION CONSTANT
C          (R*8)  CY1     = PARAMETER = EQUATION CONSTANT
C          (R*8)  CY2     = PARAMETER = EQUATION CONSTANT
C          (R*8)  CY3     = PARAMETER = EQUATION CONSTANT
C          (R*8)  CY4     = PARAMETER = EQUATION CONSTANT
C          (R*8)  C0N     = PARAMETER = EQUATION CONSTANT
C          (R*8)  C1N     = PARAMETER = EQUATION CONSTANT
C          (R*8)  C2N     = PARAMETER = EQUATION CONSTANT
C          (R*8)  C3N     = PARAMETER = EQUATION CONSTANT
C          (R*8)  C1D     = PARAMETER = EQUATION CONSTANT
C          (R*8)  C2D     = PARAMETER = EQUATION CONSTANT
C          (R*8)  C3D     = PARAMETER = EQUATION CONSTANT
C          (R*8)  C4D     = PARAMETER = EQUATION CONSTANT
C
C          (R*8)  A       = GENERAL VARIABLE
C          (R*8)  Y       = GENERAL VARIABLE
C          (R*8)  Z       = GENERAL VARIABLE
C
C ROUTINES: NONE
C
C AUTHOR:   PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C           K1/0/81
C           JET EXT. 4569
C
C DATE:     18/09/90
C
C VERSION  : 1.2
C DATE     : 20-12-2001
C MODIFIED : Martin O'Mullane
C               - Removed mainframe listing information beyond column 72.
C
C VERSION  : 1.3                          DATE: 17-4-07
```

```
C MODIFIED : HUGH SUMMERS
C            - COMPLETED COMMENT BLOCK DESCRIPTION
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      REAL*8              X
```

9.53 r8fmon1: Subroutine r8fmon1 from library adaslib

```
C
       REAL*8 FUNCTION R8FMON1(E1,E2,L)
       IMPLICIT REAL*8(A-H,O-Z)
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 FUNCTION: R8FMON1 **********************
C
C PURPOSE: CALCULATES THE MONOPOLE INTEGRAL |<E1,L|1/r>|E2,L>|^2
C
C
C NOTE: CREATED BY  ALAN BURGESS FOR USE IN THE DIPOLE INTEGRAL
C       I(KAPPA1,L1,KAPPA2,L2,1) EVALUATION AS DEFINED IN PHIL.
C       TRANS. ROY. SOC. A226,255,1970, WHERE E1=KAPPA1**2 AND
C       E2=KAPPA2**2. APPLIES TO POSITIVE ELECTRON ENERGIES, .
c       THAT IS THE FREE-FREE CASE.
C
C CALLING PROGRAMS: R8MON1
C
C INPUT:  (R*8)  E1      = KAPPA1**2 WHERE KAPPA1 IS SCALED INITIAL
C                          ELECTRON WAVE NUMBER
C INPUT:  (I*4)  L       = ORBITAL ANGULAR OMENTUM OF INITIAL ELECTRON
C INPUT:  (R*8)  E2      = KAPPA2**2 WHERE KAPPA2 IS SCALED INITIAL
C                          ELECTRON WAVE NUMBER
C
C OUTPUT: (R*8)  R8MON1 = |<E1,L|1/r>|E2,L>|^2
C
C ROUTINES:
C          ROUTINE    SOURCE   BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          ARGAM      ADAS     CALCULATES ARGGAMMA(L+1+I*A)
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 17-04-07
C MODIFIED: HUGH SUMMERS
C               - FIRST FULLY COMMENTED RELEASE
C
C-----------------------------------------------------------------------
       IF(E1+E2-1.0D-40)28,28,29
   28  R8FMON1=1.0D50
       RETURN
   29  CONTINUE
       VMAX=200.0D0
       X1=DSQRT(E1)
       X2=DSQRT(E2)
       X3=X1+X2
       X4=X3*X3
       X5=X1*X2
       X6=X2-X1
       X7=4.0D0/X4
       PI=3.141592653589793D0
       IF(E1-E2)1,1,2
    1  ETA=1.0D0/X2
       GO TO 3
    2  ETA=1.0D0/X1
    3  G=0.5D0*PI*DEXP(-PI*ETA)
       A1=1.0D0
       A2=1.0D0
       MG=0
```

```
    MA1=0
    MA2=0
    M=-1
 4  M=M+1
    EM=M
    T=EM+EM+1.0D0
    G=G*X7/(T*(T+1.0D0))
    EMM=EM*EM
    A1=A1*(1.0D0+EMM*E1)
    A2=A2*(1.0D0+EMM*E2)
30  IF(G-0.015625D0) 31,32,32
31  G=64.0D0*G
    MG=MG-1
    GO TO 30
32  IF(G-64.0D0) 34,34,33
33  G=0.015625D0*G
    MG=MG+1
    GO TO 32
34  IF(A1-64.0D0) 36,36,35
35  A1=0.015625D0*A1
    MA1=MA1+1
    GO TO 34
36  IF(A2-64.0D0) 38,38,37
37  A2=0.015625D0*A2
    MA2=MA2+1
    GO TO 36
38  CONTINUE
    IF(M-L)4,5,5
 5  G=G*(T+1.0D0)
    IF(X1-300.0D0)7,6,6
 6  B=PI/X1
    A1=1.5D0*A1/(B*(3.0D0-B*(3.0D0-B*(2.0D0-B))))
    GO TO 9
 7  IF(X1-0.2D0)9,9,8
 8  B=-PI/X1
    A1=A1/(1.0D0-DEXP(B+B))
 9  IF(X2-300.0D0)11,10,10
10  B=PI/X2
    A2=1.5D0*A2/(B*(3.0D0-B*(3.0D0-B*(2.0D0-B))))
    GO TO 13
11  IF(X2-0.2)13,13,12
12  B=-PI/X2
    A2=A2/(1.0D0-DEXP(B+B))
13  G=G*DSQRT(A1*A2)*(8.0D0)**(MG+MG+MA1+MA2)
    S0=1.0D0
    S1=0.0D0
    U=L
    V=0.0D0
    W=U+U+1.0D0
    T0=1.0D0
    T1=0.0D0
14  U=U+1.0D0
    V=V+1.0D0
    W=W+1.0D0
    IF(V-VMAX)21,21,20
20  R8FMON1=0.0D0
    RETURN
21  CONTINUE
    U0=U*U*X5+1.0D0
    U1=U*X6
    T=T0*U0-T1*U1
```

```
     T1=T0*U1+T1*U0
     T0=T
     T=X7/(V*W)
     T0=T*T0
     T1=T*T1
     S0=S0+T0
     S1=S1+T1
     S=S0*S0+S1*S1
     T=T0*T0+T1*T1
     IF(S-1.0D24*T)14,15,15
15   R8FMON1=G*DSQRT(S)
     IV=V
     RETURN
    END
    INTEGER             L
    REAL*8              E1,          E2
```

9.54 r8form: Subroutine r8form from library adaslib

```
      FUNCTION R8FORM( MXNENG , MXNSHL , N      , L      ,
     &                 IESEL  , ITYPE  , NENRGY , XLCUTA ,
     &                 PL2A   , PL3A
     &               )
C
C
C-----------------------------------------------------------------------
C
C  ******************* FORTRAN77 FUNCTION: R8FORM **********************
C
C  PURPOSE:  CALCULATES CHARGE EXCHANGE L-RESOLVED CROSS-SECTION AS A
C            FRACTION OF THE CORRESPONDING N-RESOLVED CROSS-SECTION.
C
C  CALLING PROGRAM: GENERAL USE.
C
C  FUNC  : (R*8)  R8FORM    = L-RESOLVED CROSS-SECTION AS FRACTION OF
C                             N-REOSLVED CROSS-SECTION.
C  INPUT : (I*4)  MXNENG    = MAXIMUM NO. OF ENERGIES.
C  INPUT : (I*4)  MXNSHL    = MAXIMUM NO. OF N SHELLS.
C  INPUT : (I*4)  N         = PRINCIPAL QUANTUM NUMBER.
C  INPUT : (I*4)  L         = ORBITAL QUANTUM NUMBER.
C  INPUT : (I*4)  IESEL     = SELECTED ENERGY INDEX.
C  INPUT : (I*4)  ITYPE     = TYPE OF APPROXIMATION TO USE.
C                           = 1 = 2L+1 INCREASE WITH L AND EXPONENTIAL
C                               CUTOFF AS L/LCUT.
C                           = 2 = 2L+1 INCREASE WITL L, SHARP CUTOFF AT
C                               MIN(LCUT,N-1).
C                           = 3 = (2L+1)**2 INCREASE WITH L, SHARP
C                               CUTOFF AT MIN(LCUT,N-1).
C                           = 4 = KRONECKER DELTA(L,MIN(LCUT,N-1)).
C                           = 5 = 2L+1 INCREASE WITH L AND EXP. CUTOFF
C                               AS MAX(0,L-LCUT)/2.
C                           = 6 = 2L+1 INCREASE WITH L AND EXP. CUTOFF
C                               AS 2*MAX(0,L-LCUT).
C                           = 7 = NEW PRIMARY FORM BASED ON SPFMAN13
C                               FITTING PROCEDURE WITH SHARP SWITCHING
C                               FUNCTIONS.
C                           = 8 = NEW PRIMARY FORM BASED ON SPFMAN13
C                               FITTING PROCEDURE WITH SOFT SWITCHING
C                               FUNCTIONS.
C  INPUT : (I*4)  NENRGY    = NUMBER OF ENERGIES IN DATASET.
C  INPUT : (R*8)  XLCUTA()  = PARAMETERS FOR CALCULATING L-RES X-SEC.
C                             DIMENSION: ENERGY INDEX
C  INPUT : (R*8)  PL2A()    = PARAMETERS FOR CALCULATING L-RES X-SEC.
C                             DIMENSION: ENERGY INDEX
C  INPUT : (R*8)  PL3A()    = PARAMETERS FOR CALCULATING L-RES X-SEC.
C                             DIMENSION: ENERGY INDEX
C
C  PARAM : (I*4)  MXB       = 'MXBEAM'.
C  PARAM : (I*4)  MXN       = 'MXNSHL'.
C  PARAM : (I*4)  MXTYPE    = NO. OF DIFFERENT APPROXIMATIONS.
C  PARAM : (R*8)  C1        =
C
C          (I*4)  LCUT      = CUT OFF VALUE FOR ORBITAL QUANTUM NUMBER.
C          (I*4)  LC        = CUT OFF VALUE FOR ORBITAL QUANTUM NUMBER.
C          (I*4)  NN        = PRINCIPAL QUANTUM NUMBER LOOP INDEX.
C          (I*4)  LL        = ORBITAL QUANTUM NUMBER LOOP INDEX.
C          (I*4)  IE        = ENERGY LOOP INDEX.
C
```

```
C          (R*8)  SUM       =
C          (R*8)  XLC       = REAL VALUE = LC.
C          (R*8)  XL        = REAL VALUE = L.
C          (R*8)  XLL       = REAL VALUE = LL.
C          (R*8)  EF        =
C          (R*8)  XLCRIT    =
C          (R*8)  T         =
C          (R*8)  S1        =
C          (R*8)  S2        =
C          (R*8)  T1        =
C          (R*8)  T2        =
C
C          (R*8)  SUM1A()   = TABLE OF SUMS FOR 1ST APPROXIMATION.
C                             1ST DIMENSION: L CUTOFF
C                             2ND DIMENSION: N-SHELL
C          (R*8)  SUM5A()   = TABLE OF SUMS FOR 5TH APPROXIMATION.
C                             1ST DIMENSION: L CUTOFF
C                             2ND DIMENSION: N-SHELL
C          (R*8)  SUM6A()   = TABLE OF SUMS FOR 6TH APPROXIMATION.
C                             1ST DIMENSION: L CUTOFF
C                             2ND DIMENSION: N-SHELL
C          (R*8)  SUM7A()   = TABLE OF SUMS FOR 7TH APPROXIMATION.
C                             1ST DIMENSION: ENERGY INDEX
C                             2ND DIMENSION: N-SHELL
C          (R*8)  SUM8A()   = TABLE OF SUMS FOR 8TH APPROXIMATION.
C                             1ST DIMENSION: ENERGY INDEX
C                             2ND DIMENSION: N-SHELL
C
C          (L*4)  LFIRST()  = FLAGS IF FIRST CALL OF APPROXIMATION. ONLY
C                             USED BY APPROX. WHICH REQUIRE AN INTIAL
C                             SUM
C                           = .TRUE.  = FIRST CALL.
C                           = .FALSE. = NOT FIRST CALL.
C                             1ST DIMENSION: APPROX. TYPE INDEX
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4UNIT     ADAS      RETURN UNIT NO. FOR OUTPUT OF MESSAGES.
C
C AUTHOR:  JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 5183
C
C DATE:    19/10/93
C
C UPDATES:
C
C VERSION: 1.2 DATE: 23-06-98
C MODIFIED: RICHARD MARTIN
C - INCREASED MXE FORM 30 TO 40.
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IESEL,       ITYPE,       L,           MXNENG
      INTEGER             MXNSHL,      N,           NENRGY
      REAL*8              PL2A(MXNENG),             PL3A(MXNENG)
      REAL*8              XLCUTA(MXNENG)
```

9.55 r8fun1: Subroutine r8fun1 from library adaslib

```
      FUNCTION R8FUN1 ( Z )
C-----------------------------------------------------------------------
C
C  ***************** FORTRAN77 REAL*8 FUNCTION: R8FUN1 ******************
C
C  PURPOSE: Returns argument
C
C  CALLING PROGRAM: GENERAL USE
C
C  FUNCTION:
C           (R*8)  R8FUN1  = FUNCTION NAME
C           (R*8)  Z       = INPUT VALUE
C
C AUTHOR:   PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C           K1/0/81
C           JET EXT. 4569
C
C DATE:     13/08/90
C
C VERSION  : 1.2
C DATE     : 20-12-2001
C MODIFIED : Martin O'Mullane
C               - Removed mainframe listing information beyond column 72.
C
C VERSION  : 1.3
C DATE     : 10-04-2007
C MODIFIED : Allan Whiteford
C               - Modified documentation as part of automated
C   subroutine documentation preparation.
C
C-----------------------------------------------------------------------
      REAL*8              Z
```

9.56 r8fun2: Subroutine r8fun2 from library adaslib

```
      FUNCTION R8FUN2 ( Z )
C-----------------------------------------------------------------------
C
C  ***************** FORTRAN77 REAL*8 FUNCTION: R8FUN2 ******************
C
C  PURPOSE: RETURN 1 / ( Z+1 )
C
C  CALLING PROGRAM: GENERAL USE
C
C  FUNCTION:
C           (R*8)  R8FUN2  = FUNCTION NAME
C           (R*8)  Z       = INPUT VALUE
C
C AUTHOR:   PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C           K1/0/81
C           JET EXT. 4569
C
C DATE:     13/08/90
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 06-09-95
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST RELEASE
C
C VERSION  : 1.2
C DATE     : 20-12-2001
C MODIFIED : Martin O'Mullane
C               - Removed mainframe listing information beyond column 72.
C
C VERSION  : 1.3
C DATE     : 10-04-2007
C MODIFIED : Allan Whiteford
C               - Modified documentation as part of automated
C   subroutine documentation preparation.
C
C-----------------------------------------------------------------------
      REAL*8              Z
```

9.57 r8gam: Subroutine r8gam from library adaslib

```
      FUNCTION R8GAM( INDEX )
C
C
C-----------------------------------------------------------------------
C
C  ***************** FORTRAN77 INTEGER FUNCTION: R8GAM ******************
C
C  PURPOSE: USES INDEX TO REFERENCE 'GAM' TABLE GENERATED BY SUBROUTINE
C           'XXGAMA'.
C
C           THE 'JGAM' TABLE IS REFERENCED WITH THE FUNCTION 'I4JGAM'.
C
C
C  CALLING PROGRAM: GENERAL USE.
C
C  FUNC  : (I*4)  I4GAM   =
C
C  INPUT : (I*4)  INDEX   =
C
C  PARAM : (I*4)  MXINDX  = 200
C
C          (L*4)  LFIRST   = .TRUE.  = FIRST TIME FUNCTION CALLED.
C                            .FLASE. = FUNCTION HAS BEEN CALLED BEFORE.
C
C          (I*4)  JGAM()  =
C                           DIMENSION: REFERENCED BY 'INDEX'.
C          (I*4)  GAM()   =
C                           DIMENSION: REFERENCED BY 'INDEX'.
C
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4UNIT     ADAS      RETURNS UNIT NO. FOR OUPUT OF MESSAGES.
C          XXGAMA     ADAS      FILLS 'GAM'.
C
C AUTHOR:   JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C           K1/0/81
C           JET EXT. 5183
C
C DATE:     30/09/93
C
C VERSION  : 1.2
C DATE     : 20-12-2001
C MODIFIED : Martin O'Mullane
C               - Removed mainframe listing information beyond column 72.
C
C VERSION  : 1.3
C DATE     : 10-04-2007
C MODIFIED : Allan Whiteford
C               - Modified documentation as part of automated
C   subroutine documentation preparation.
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             INDEX
```

9.58 r8gav: Subroutine r8gav from library adaslib

```
      function r8gav(u, gam2, n0)

C-----------------------------------------------------------------------
C
C  ***************** FORTRAN77 REAL*8 FUNCTION: R8GAV ******************
C
C  PURPOSE: CALCULATES TOTAL GAUNT FACTOR FOR FREE-FREE AND
C  QUASI-CONTINUOUS FREE-BOUND TRANSITIONS
C
C  INPUT
C      U=HV/KT  WHERE HV IS PHOTON ENERGY
C               AND KT IS ELECTRON TEMPERATURE (ENERGY UNITS)
C      GAM2=Z*Z*IH/KT  WHERE Z IS TARGET ION CHARGE
C                      AND IH IS THE RYDBERG ENERGY
C      N0       IS THE FIRST BOUND LEVEL TO BE INCLUDED IN THE
C               INTEGRAL
C  OUTPUT
C      GAV=MAXWELLIAN AVERAGED GAUNT FACTOR.
C
C
C  AUTHOR   : L D Horton
C  DATE     : 24-06-1997
C
C
C
C  VERSION  : 1.1
C  DATE     : 02-03-2005
C  MODIFIED : Martin O'Mullane
C                - First version in central ADAS.
C                - Make implicit none.
C
C  VERSION  : 1.2
C  DATE     : 15-03-2006
C  MODIFIED : Allan Whiteford
C                - Removed external giii statement.
C
C  VERSION  : 1.3
C  DATE     : 22-06-2006
C  MODIFIED : Martin O'Mullane
C                - Removed redundant external gbf statement.
C
C  VERSION  : 1.4
C  DATE     : 10-04-2007
C  MODIFIED : Allan Whiteford
C               - Modified documentation as part of automated
C   subroutine documentation preparation.
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             N0
      REAL*8              GAM2,        U
```

9.59 r8gbf: Subroutine r8gbf from library adaslib

```
      FUNCTION R8GBF( EN , U )
C
C
C-----------------------------------------------------------------------
C
C  ***************** FORTRAN77 REAL*8 FUNCTION: R8GBF  ******************
C
C  PURPOSE: CALCULATES BOUND-FREE GAUNT FACTORS
C
C  CALLING PROGRAM: GENERAL USE
C
C  FUNC  : (R*8)  R8GBF   =
C
C  INPUT : (R*8)  EN      =
C  INPUT : (R*8)  U       = N**2 / K**2 = N**2 * E / Z**2
C
C  PARAM : (R*8)  P1      = 2/3.
C  PARAM : (R*8)  P2      = 25/18.
C  PARAM : (R*8)  P3      = 4/3
C
C          (R*8)  X       =
C          (R*8)  T1      =
C          (R*8)  T2      =
C
C ROUTINES: NONE
C
C AUTHOR:  JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 5183
C
C DATE:    04/11/93
C
C UNIX-IDL PORT:
C
C AUTHOR:  WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C
C DATE:    22ND MAY 1996
C
C VERSION: 1.1                          DATE: 22-05-96
C MODIFIED: WILLIAM OSBORN
C               - FIRST VERSION. IBM VERSION NOT CHANGED
C
C VERSION : 1.2
C DATE    : 02-03-2005
C MODIFIED: Martin O'Mullane
C            - Change P2 from 25/18 to 28/18 as given by equation B3
C              in Burgess & Summers, MNRAS, vol 174, (1976), p345.
C
C VERSION  : 1.3
C DATE     : 10-04-2007
C MODIFIED : Allan Whiteford
C               - Modified documentation as part of automated
C   subroutine documentation preparation.
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      REAL*8              EN,          U
```

9.60 r8giiav: Subroutine r8giiav from library adaslib

```
      function r8giiav(u, u2, n0)

C-----------------------------------------------------------------------
C
C  *************** FORTRAN77 REAL*8 FUNCTION: R8GIIAV *****************
C
C  PURPOSE: CALCULATE AVERAGED FREE-BOUND GAUNT FACTOR FOR SUMMED
C  CONTRIBUTIONS OF THE HIGH LYING STATES - SUMMERS AND HOOPER EQU. 19
C
C  INPUT
C      U=HV/IH  WHERE HV IS PHOTON ENERGY
C               AND IH IS THE RYDBERG ENERGY
C      U2=K*TE/IH  WHERE TE IS THE ELECTRON TEMPERATURE
C                  AND K IS THE BOLTZMANN CONSTANT
C      N0       IS THE FIRST LEVEL TO BE COUNTED IN THIS
C               QUASI-CONTINUUM
C  OUTPUT
C      GIIAV=AVERAGED FREE-BOUND GAUNT FACTOR.
C
C
C  AUTHOR   : L D Horton
C  DATE     : 24-06-1997
C
C
C  VERSION  : 1.1
C  DATE     : 02-03-2005
C  MODIFIED : Martin O'Mullane
C                - First version in central ADAS.
C
C  VERSION  : 1.3
C  DATE     : 10-04-2007
C  MODIFIED : Allan Whiteford
C               - Modified documentation as part of automated
C   subroutine documentation preparation.
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             N0
      REAL*8              U,           U2
```

9.61 r8giih: Subroutine r8giih from library adaslib

```
      FUNCTION R8GIIH ( VVE    , V      , N      , L      ,
     &                  L1     , LP     , ISP    , LT     ,
     &                  LT1    , IS     , IRES
     &                )
C
C
C-----------------------------------------------------------------------
C
C  ***************** FORTRAN77 REAL*8 FUNCTION: R8GIIH ******************
C
C  PURPOSE:  CALCULATES BOUND-FREE G-FACTORS FOR ANGULARLY RESOLVED
C            LEVELS USES HYDROGENIC MATRIX ELEMENTS.
C
C            FOR COMPLETENESS, THE UNRESOLVED, BUNDLED N, GBF (BURGESS
C            AND SUMMERS ,1976) CAN ALSO BE OBTAINED.
C
C  CALLING PROGRAM: GENERAL USE
C
C  FUNC  : (R*8)  R8GIIH  = BOUND-FREE GAUNT FACTOR
C
C  INPUT : (R*8)  VVE     = V**2*E WHERE E=(FREE ELECTRON ENERGY)/Z**2.
C                           UNITS: RYD
C  INPUT : (R*8)  V       = EFFECTIVE PRINCIPAL QUANTUM NUMBER OF BOUND
C                           ELECTRON.
C  INPUT : (I*4)  N       = PRINCIPAL QUANTUM NUMBER OF BOUND ELECTRON.
C  INPUT : (I*4)  L       = ORBITAL QUANTUM NUMBER OF BOUND ELECTRON.
C  INPUT : (I*4)  L1      = ORBITAL QUANTUM NUMBER OF FREE ELECTRON.
C  INPUT : (I*4)  LP      = TOTAL ORBITAL ANGULAR MOMENTUM QUANTUM
C                           NUMBER OF PARENT STATE.
C  INPUT : (I*4)  ISP     = 2*SP+1 WHERE SP IS TOTAL SPIN OF PARENT
C                           STATE.
C  INPUT : (I*4)  LT      = TOTAL ORBITAL ANGULAR MOMENTUM QUANTUM
C                           NUMBER OF BOUND SYSTEM.
C  INPUT : (I*4)  LT1     = TOTAL ORBITAL ANGULAR MOMENTUM QUANTUM
C                           NUMBER OF FREE SYSTEM.
C  INPUT : (I*4)  IS      = 2*S+1 WHERE S IS TOTAL SPIN OF SYSTEM.
C  INPUT : (I*4)  IRES    = LEVEL OF RESOLUTION.
C                         = 1 :
C                         = 2 : ABOVE LT1 SUM.
C                         = 3 : ABOVE LT SUM.
C                         = 4 : ABOVE S SUM.
C                         = 5 : UNRESOLVED GBF.
C
C          (R*8)  XN      = REAL VALUE = N.
C          (R*8)  XL      = REAL VALUE = L.
C          (R*8)  XL1     = REAL VALUE = L1.
C          (R*8)  XLP     = REAL VALUE = LP.
C          (R*8)  XSP     = REAL VALUE = ISP.
C          (R*8)  XLT     = REAL VALUE = LT.
C          (R*8)  XLT1    = REAL VALUE = LT1.
C          (R*8)  XS      = REAL VALUE = IS.
C          (R*8)  W       =
C          (R*8)  ANG     =
C          (R*8)  CL1     =
C          (R*8)  E2      = -E WHERE E=(FREE ELECTRON ENERGY)/Z**2.
C                           UNITS: RYD
C          (R*8)  THETH   =
C
C ROUTINES:
```

```
C           ROUTINE    SOURCE    BRIEF DESCRIPTION
C           ------------------------------------------------------------
C           R8WIG6     ADAS
C           R8RD2F     ADAS      RETURNS HYDROGENIC BOUND-FREE RADIAL
C                                INTEGRALS.
C           R8GBF      ADAS
C
C AUTHOR:  JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 5183
C
C DATE:    04/11/93
C
C UNIX-IDL PORT:
C
C AUTHOR:  WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C
C DATE:    22ND MAY 1996
C
C VERSION: 1.1                          DATE: 22-05-96
C MODIFIED: WILLIAM OSBORN
C               - FIRST VERSION. IBM VERSION NOT CHANGED
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IRES,        IS,          ISP,         L
      INTEGER             L1,          LP,          LT,          LT1
      INTEGER             N
      REAL*8              V,           VVE
```

9.62 r8giii: Subroutine r8giii from library adaslib

```
      function r8giii(jz, l, e1, e2)

C-----------------------------------------------------------------------
C
C  ***************** FORTRAN77 REAL*8 FUNCTION: R8GIII *****************
C
C  PURPOSE: CALCULATES GIII GIVEN IN EQUATIONS (11) AND (15) OF
C  A. BURGESS, J. PHYS. B7, L364, 1974.
C
C  SET L=1.SET JZ ZERO FOR THE ZERO CHARGE (NEUTRAL ATOM) CASE.
C  FOR PARTIAL SUMS SET L TO LOWER LIMIT OF SUMMATION WHICH
C  MUST BE GREATER THAN ZERO.
C  SET E1=(KAPPA1)**2 FOR NON ZERO CHARGE,
C        =(K1)**2 FOR ZERO CHARGE.
C  SET E2=(KAPPA2)**2 FOR NON ZERO CHARGE,
C        =(K2)**2 FOR ZERO CHARGE.
C
C
C  AUTHOR   : H P Summers
C  DATE     : 22-11-1984
C
C
C
C  VERSION  : 1.1
C  DATE     : 02-03-2005
C  MODIFIED : Martin O'Mullane
C                - First version in central ADAS.
C                - Make implicit none.
C  VERSION  : 1.2
C  DATE     : 16-05-2005
C  MODIFIED : Allan Whiteford
C                - Call r8fdip and r8fdip0 instead of fdip and fdip0
C                  since the routines were renamed.
C
C  VERSION  : 1.3
C  DATE     : 10-04-2007
C  MODIFIED : Allan Whiteford
C               - Modified documentation as part of automated
C   subroutine documentation preparation.
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             JZ,          L
      REAL*8              E1,          E2
```

9.63 r8giiiav: Subroutine r8giiiav from library adaslib

```
      function r8giiiav(u, gam2)

C-----------------------------------------------------------------------
C
C  *************** FORTRAN77 REAL*8 FUNCTION: R8GIIIAV ***************
C
C
C  PURPOSE: CALCULATES MAXWELLIAN AVERAGED FREE-FREE GAUNT FACTORS
C
C  INPUT
C      U=HV/KT  WHERE HV IS PHOTON ENERGY
C               AND KT IS ELECTRON TEMPERATURE (ENERGY UNITS)
C      GAM2=Z*Z*IH/KT  WHERE Z IS TARGET ION CHARGE
C               AND IH IS THE RYDBERG ENERGY
C  OUTPUT
C      GIIIAV=MAXWELLIAN AVERAGED FREE-FREE GAUNT FACTOR.
C
C
C  AUTHOR   : H P Summers
C  DATE     : 22-11-1984
C
C
C  VERSION  : 1.1
C  DATE     : 02-03-2005
C  MODIFIED : Martin O'Mullane
C               - First version in central ADAS.
C               - Make implicit none.
C
C  VERSION  : 1.2
C  DATE     : 15-03-2006
C  MODIFIED : Allan Whiteford
C               - Removed external giii statement
C
C  VERSION  : 1.3
C  DATE     : 10-04-2007
C  MODIFIED : Allan Whiteford
C              - Modified documentation as part of automated
C   subroutine documentation preparation.
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      REAL*8              GAM2,        U
```

9.64 r8p: Subroutine r8p from library adaslib

```
      FUNCTION R8P ( N , L )
C
C
C-----------------------------------------------------------------------
C
C  ***************** FORTRAN77 REAL*8 FUNCTION: R8P    ******************
C
C  PURPOSE: EXPECTATION VALUE <1/R^4> IN HYDROGEN APPROXIMATION FOR USE
C           IN DIPOLE POLARISABILITY ENERGY SHIFT EVALUATION
C
C  CALLING PROGRAM: CXLTHE, CXSODE
C
C  FUNC:   (R*8)  R8P     = EXPECTATION VALUE <1/R^4>
C
C  INPUT:  (I*4)  N       = PRINCIPAL QUANTUM NUMBER.
C  INPUT:  (I*4)  L       = ORBITAL QUANTUM NUMBER.
C
C          (R*8)  XN      = REAL VALUE = N.
C          (R*8)  XL      = REAL VALUE = L.
C
C ROUTINES: NONE
C
C AUTHOR:  JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 5183
C
C DATE:    02/11/93
C
C VERSION: 1.1                          DATE: 02-11-093
C MODIFIED: JONATHAN NASH
C               - FIRST RELEASE
C
C VERSION: 1.2                          DATE: 17-04-07
C MODIFIED: HUGH SUMMERS
C               - FIRST FULLY COMMENTED RELEASE
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             L,           N
```

9.65 r8prov: Subroutine r8prov from library adaslib

```
      FUNCTION R8PROV ( N1 , L1 , N2 , L2 )
C
C
C-----------------------------------------------------------------------
C
C  ******************* FORTRAN77 FUNCTION: R8PROV **********************
C
C  PURPOSE:  APPROXIMATION TO HYDROGENIC OVERLAP INTEGRAL FOR CLASSICAL
C            BINARY ENCOUNTER SPIN CHANGE CROSS-SECTIONS FOR
c            MAX0(N,N1) LARGE.
C
C  NOTE   : SEE OVLP FOR THE GENERAL CASE
C
C  CALLING PROGRAM: GENERAL
C
C  FUNC  : (R*8)  R8PROV  = OVERLAP INTEGRAL
C
C  INPUT : (I*4)  N1      = FIRST N QUANTUM NUMBER.
C  INPUT : (I*4)  L1      = FIRST L QUANTUM NUMBER.
C  INPUT : (I*4)  N2      = SECOND N QUANTUM NUMBER.
C  INPUT : (I*4)  L2      = SECOND L QUANTUM NUMBER.
C
C  PARAM : (R*8)  P1      = 0.3
C
C          (I*4)  M       =
C
C          (R*8)  XN1     = REAL VALUE = N1.
C          (R*8)  XL1     = REAL VALUE = L1.
C          (R*8)  XN2     = REAL VALUE = N2.
C          (R*8)  XL2     = REAL VALUE = L2.
C          (R*8)  XM      = REAL VALUE = M.
C          (R*8)  BT      =
C          (R*8)  X1      =
C          (R*8)  X2      =
C          (R*8)  XK      =
C
C ROUTINES: NONE
C
C AUTHOR:   JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C           K1/0/81
C           JET EXT. 5183
C
C DATE:     04/10/93
C
C VERSION: 1.1                          DATE: 02-11-093
C MODIFIED: JONATHAN NASH
C               - FIRST RELEASE
C
C VERSION: 1.2                          DATE: 17-04-07
C MODIFIED: HUGH SUMMERS
C               - FIRST FULLY COMMENTED RELEASE
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             L1,          L2,          N1,          N2
```

9.66 r8qp: Subroutine r8qp from library adaslib

```
      FUNCTION R8QP ( N , L )
C
C
C-----------------------------------------------------------------------
C
C  ***************** FORTRAN77 REAL*8 FUNCTION: R8QP   ******************
C
C  PURPOSE: EXPECTATION VALUE <1/R^6> IN HYDROGEN APPROXIMATION FOR USE
c           IN QUADRUPOLE POLARISABILITY ENERGY SHIFT EVALUATION
C
C  CALLING PROGRAM: CXLTHE, CXSODE
C
C
C  CALLING PROGRAM: GENERAL USE
C
C  FUNC:   (R*8)  R8QP    = EXPECTATION VALUE <1/R^6>
C
C  INPUT:  (I*4)  N       = PRINCIPAL QUANTUM NUMBER.
C  INPUT:  (I*4)  L       = ORBITAL QUANTUM NUMBER.
C
C          (R*8)  XN      = REAL VALUE = N.
C          (R*8)  XL      = REAL VALUE = L.
C
C ROUTINES: NONE
C
C AUTHOR:  JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 5183
C
C DATE:    02/11/93
C
C VERSION: 1.1                          DATE: 02-11-93
C MODIFIED: JONATHAN NASH
C               - FIRST RELEASE
C
C VERSION: 1.2                          DATE: 17-04-07
C MODIFIED: HUGH SUMMERS
C               - FIRST FULLY COMMENTED RELEASE
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             L,           N
```

9.67 r8rd2b: Subroutine r8rd2b from library adaslib

```
      FUNCTION R8RD2B ( NU , LU , NL , LL )
C
C
C-----------------------------------------------------------------------
C
C  ***************** FORTRAN77 REAL*8 FUNCTION: R8RD2B ******************
C
C  PURPOSE: CLACULATES HYDROGENIC BOUND-BOUND RADIAL INTEGRALS USING
C           RECURRENCE RELATIONS.
C
C  CALLING PROGRAM: GENERAL USE
C
C  FUNC:   (R*8)  R8RD2B  = RESULT OF INTEGRAL.
C
C  INPUT:  (I*4)  NU      = UPPER VALUE OF N QUANTUM NUMBER.
C  INPUT:  (I*4)  LU      = UPPER VALUE OF L QUANTUM NUMBER.
C  INPUT:  (I*4)  NL      = LOWER VALUE OF N QUANTUM NUMBER.
C  INPUT:  (I*4)  LL      = LOWER VALUE OF L QUANTUM NUMBER.
C
C  PARAM:  (R*8)  P1      = EQUATION CONSTANT.
C  PARAM:  (R*8)  P12     = EQUATION CONSTANT = P1**2.
C  PARAM:  (R*8)  P2      = EQUATION CONSTANT = 1/P1.
C  PARAM:  (R*8)  P22     = EQUATION CONSTANT = P2**2.
C
C          (I*4)  JS      =
C          (I*4)  I       = LOOP INDEX.
C
C          (R*8)  XNL     = REAL VALUE = NL.
C          (R*8)  XNL2    = REAL VALUE = XNL**2.
C          (R*8)  XNU     = REAL VALUE = NU.
C          (R*8)  XNU2    = REAL VALUE = XNU**2.
C          (R*8)  XNL1    = REAL VALUE = NL-1.
C          (R*8)  U       =
C          (R*8)  V       =
C          (R*8)  W       =
C          (R*8)  P       =
C          (R*8)  XI      = REAL VALUE = I.
C          (R*8)  XI1     = REAL VALUE = I+1.
C          (R*8)  XI12    = REAL VALUE = XI1**2.
C          (R*8)  T1      =
C          (R*8)  T2      =
C          (R*8)  T3      =
C
C ROUTINES: NONE
C
C AUTHOR:   JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C           K1/0/81
C           JET EXT. 5183
C
C DATE:     28/09/93
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             LL,          LU,          NL,          NU
```

9.68 r8rd2f: Subroutine r8rd2f from library adaslib

```
       FUNCTION R8RD2F( N , L , L1 , E2 )
C
C
C-----------------------------------------------------------------------
C
C  ***************** FORTRAN77 REAL*8 FUNCTION: R8RD2F ******************
C
C  PURPOSE: CLACULATES HYDROGENIC BOUND-FREE RADIAL INTEGRALS USING
C           RECURRENCE RELATIONS.
C
C  CALLING PROGRAM: GENERAL USE
C
C  FUNC:   (R*8)  R8RD2F  = RESULT OF INTEGRAL.
C
C  INPUT:  (I*4)  N       = PRINCIAPL QUANTUM NUMBER OF BOUND ELECTRON.
C  INPUT:  (I*4)  L       = ORBITAL QUANTUM NUMBER OF BOUND ELECTRON.
C  INPUT:  (I*4)  L1      = ORBITAL QUANTUM NUMBER OF FREE ELECTRON.
C  INPUT:  (R*8)  E2      =
C
C  PARAM:  (R*8)  P1      = EQUATION CONSTANT = 64.
C  PARAM:  (R*8)  P12     = EQUATION CONSTANT = P1**2.
C  PARAM:  (R*8)  P2      = EQUATION CONSTANT = 1/P1.
C  PARAM:  (R*8)  P22     = EQUATION CONSTANT = P2**2.
C
C          (I*4)  JS      =
C          (I*4)  I       = LOOP INDEX.
C
C          (R*8)  XN      = REAL VALUE = N.
C          (R*8)  XN1     = REAL VALUE = N-1.
C          (R*8)  XN2     = REAL VALUE = XN**2.
C          (R*8)  E       =
C          (R*8)  U       =
C          (R*8)  V       =
C          (R*8)  P       =
C          (R*8)  XI      = REAL VALUE = I.
C          (R*8)  XI1     = REAL VALUE = I+1.
C          (R*8)  XI12    = REAL VALUE = XI1**2.
C          (R*8)  T1      =
C          (R*8)  T2      =
C          (R*8)  T3      =
C
C ROUTINES: NONE
C
C AUTHOR:   JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C           K1/0/81
C           JET EXT. 5183
C
C DATE:     04/11/93
C
C UNIX-IDL PORT:
C
C AUTHOR:  WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C
C DATE:    22ND MAY 1996
C
C VERSION: 1.1                          DATE: 22-05-96
C MODIFIED: WILLIAM OSBORN
C               - FIRST VERSION. IBM VERSION NOT CHANGED
C
```

```
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             L,           L1,          N
      REAL*8              E2
```

9.69 r8scon: Subroutine r8scon from library adaslib

```
      FUNCTION R8SCON( INTYP, OUTTYP, XSIN )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: R8SCON *********************
C
C  PURPOSE: TO CONVERT AN ARRAY OF CROSS-SECTIONS INTO A SPECIFIED FORM.
C           (DOUBLE PRECISION FUNCTION VERSION OF 'XXSCON')
C
C  CALLING PROGRAM: GENERAL USE
C
C  FUNCTION:
C
C             (R*8)  R8SCON  = FUNCTION NAME -
C                              OUTPUT CROSS-SECTION (STATED UNITS)
C             (I*4)  INTYP   = 1 => 'XSIN' UNITS: CM**2
C                              = 2 => 'XSIN' UNITS: PI*(A0**2)
C             (I*4)  OUTTYP  = 1 => 'R8SCON' UNITS: CM**2
C                              = 2 => 'R8SCON' UNITS: PI*(A0**2)
C             (R*8)  XSIN    = INPUT  CROSS-SECTION (STATED UNITS)
C
C             (R*8)  A0      = PARAMETER: BOHR RADIUS = 5.29177D-09 cm
C             (R*8)  PI      = PARAMETER: Pi = 3.1415926536
C             (R*8)  CM2A0   = PARAMETER: CM**2 TO PI*(A0**2) CONVERSION
C                                         FACTOR.
C             (R*8)  A02CM   = PARAMETER: PI*(A0**2) TO CM**2 CONVERSION
C                                         FACTOR.
C
C             (R*8)  SCONV() = ENERGY/VELOCITY CONVERSION PARAMETERS
C
C ROUTINES:  NONE
C
C NOTE:
C            ENERGY/VELOCITY CONVERSION PARAMETERS:
C
C            INTYP = 1 ; SCONV(1) =>  VELOCITY: CM**2     -> OUTPUT FORM
C            INTYP = 2 ; SCONV(2) =>  VELOCITY: PI*(A0**2)-> OUTPUT FORM
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 4569
C
C DATE:    05/02/91
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             INTYP,       OUTTYP
      REAL*8              XSIN
```

9.70 r8tcon: Subroutine r8tcon from library adaslib

```
      FUNCTION R8TCON( INTYP , OUTTYP , IZ1 , TIN )
C-----------------------------------------------------------------------
C
C  ***************** FORTRAN77 REAL*8 FUNCTION: R8TCON ******************
C
C  PURPOSE: TO CONVERT A TEMPERATURE INTO SPECIFIED UNITS
C           (DOUBLE PRESCISION FUNCTION VERSION OF 'XXTCON')
C
C  CALLING PROGRAM: GENERAL USE
C
C  FUNCTION:
C
C             (R*8)  R8TCON  = FUNCTION NAME -
C                              OUTPUT TEMPERATURE (STATED UNITS)
C             (I*4)  INTYP   = 1 => 'TIN' UNITS: KELVIN
C                            = 2 => 'TIN' UNITS: eV
C                            = 3 => 'TIN' UNITS: REDUCED TEMP.
C             (I*4)  OUTTYP  = 1 => 'R8TCON' UNITS: KELVIN
C                            = 2 => 'R8TCON' UNITS: eV
C                            = 3 => 'R8TCON' UNITS: REDUCED TEMP.
C             (I*4)  IZ1     = RECOMBINING ION CHARGE (= Z+1).
C             (R*8)  TIN     = INPUT TEMPERATURE (STATED UNITS)
C
C             (R*8)  EV2KEL  = ELECTRON VOLTS TO KELVIN CONVERSION
C             (R*8)  KEL2EV  = KELVIN TO ELECTRON VOLTS CONVERSION
C
C             (R*8)  Z1P2    = 'IZ1'**2
C             (R*8)  TCONV() = TEMPERATURE CONVERSION PARAMETERS
C
C ROUTINES:  NONE
C
C NOTE:
C            TEMPERATURE CONVERSION PARAMETERS:
C
C            INTYP = 1 ; TCONV(1) =>             KELVIN  -> OUTPUT UNITS
C            INTYP = 2 ; TCONV(2) =>                 eV  -> OUTPUT UNITS
C            INTYP = 3 ; TCONV(3) => REDUCED TEMPERATURE -> OUTPUT UNITS
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 4569
C
C DATE:    04/01/91
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             INTYP,       IZ1,         OUTTYP
      REAL*8              TIN
```

9.71 r8xip:]

Subroutine r8xip from library adaslib

```
      FUNCTION R8XIP( XI , DELTA )
C
C
C-----------------------------------------------------------------------
C
C  ***************** FORTRAN77 REAL*8 FUNCTION: R8XIP  ******************
C
C  PURPOSE: EVALUATES IMPACT PARAMETER CROSS-SECTION FIRST BESSEL
C           INTEGRAL X [BURGESS AND SUMMERS: MNRAS (1976)
C           172,345 - EQN C12]
C
C  CALLING PROGRAMS: EIQIP, ZERO1, CXEIQP,CXZERO
C
C  INPUT:  (R*8)  XI      = Z[1/KN-1/KN1]/A0 WITH Z TARGET CHARGE,
C                           KN, KN1 INITIAL AND FINAL ELECTRON WAVE
C                           NUMBERS AND A0 THE BOHR RADIUS
C  INPUT:  (R*8)  DELTA   = RC[KN-KN1] WITH RC THE CLOSEST APPROACH
c
C  OUTPUT: (R*8)  R8XIP   = X(XI,DELTA)
C
C ROUTINES: NONE
C
C AUTHOR:  JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 5183
C
C DATE:    07/10/93
C
C VERSION 1.1                              DATE: 07/10/93
C MODIFIED: JONATHAN NASH
C           - FIRST VERSION
C VERSION 1.2                              DATE: 17-04-07
C MODIFIED: HUGH SUMMERS
C           - COMPLETED COMMENT BLOCK DESCRIPTION
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      REAL*8              DELTA,       XI
```

9.72 r8yip:]

Subroutine r8yip from library adaslib

```
      FUNCTION R8YIP( XI , DELTA )
C
C
C-----------------------------------------------------------------------
C
C  ***************** FORTRAN77 REAL*8 FUNCTION: R8YIP  ******************
C
C  PURPOSE: EVALUATES IMPACT PARAMETER CROSS-SECTION SECOND BESSEL
C           INTEGRAL Y [BURGESS AND SUMMERS: MNRAS (1976)
C           172,345 - EQN C14]
C
C  CALLING PROGRAMS: EIQIP, CXEIQP
C
C  INPUT:  (R*8)  XI      = Z[1/KN-1/KN1]/A0 WITH Z TARGET CHARGE,
C                           KN, KN1 INITIAL AND FINAL ELECTRON WAVE
C                           NUMBERS AND A0 THE BOHR RADIUS
C  INPUT:  (R*8)  DELTA   = RC[KN-KN1] WITH RC THE CLOSEST APPROACH
c
C  OUTPUT: (R*8)  R8YIP   = Y(XI,DELTA)
C
C ROUTINES: NONE
C
C AUTHOR:  JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 5183
C
C DATE:    07/10/93
C
C VERSION 1.1                              DATE: 07/10/93
C MODIFIED: JONATHAN NASH
C           - FIRST VERSION
C VERSION 1.2                              DATE: 17-04-07
C MODIFIED: HUGH SUMMERS
C           - COMPLETED COMMENT BLOCK DESCRIPTION
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      REAL*8              DELTA,       XI
```

9.73 r8zeta: Subroutine r8zeta from library adaslib

```
      FUNCTION R8ZETA ( ZEFF , N , L )
C
C
C-----------------------------------------------------------------------
C
C  ***************** FORTRAN77 REAL*8 FUNCTION: R8ZETA ******************
C
C  PURPOSE: HYDROGENIC SPIN-ORBIT INTERACTION ENERGY ZETA
C
C  CALLING PROGRAM: GENERAL USE
C
C  FUNC:   (R*8)  R8ZETA  = ZETA
C
C  INPUT:  (I*4)  ZEFF    = EFFECTIVE ION CHARGE.
C  INPUT:  (I*4)  N       = PRINCIPAL QUANTUM NUMBER.
C  INPUT:  (I*4)  L       = ORBITAL QUANTUM NUMBER.
C
C          (R*8)  XN      = REAL VALUE = N.
C          (R*8)  XL      = REAL VALUE = L.
C
C ROUTINES: NONE
C
C AUTHOR:  JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 5183
C
C DATE:    11/10/93
C
C VERSION: 1.1                          DATE: 11-10-93
C MODIFIED: JONATHAN NASH
C               - FIRST RELEASE
C
C VERSION: 1.2                          DATE: 17-04-07
C MODIFIED: HUGH SUMMERS
C               - FIRST FULLY COMMENTED RELEASE
C
C-----------------------------------------------------------------------
      INTEGER             L,           N
      REAL*8              ZEFF
```

9.74 xfelem: Subroutine xfelem from library adaslib

```
      FUNCTION XFELEM ( IZ0 )
C-----------------------------------------------------------------------
C
C ************* FORTRAN77 CHARACTER*12 FUNCTION: XFELEM ****************
C
C PURPOSE: TO RETURN THE NAME OF THE ELEMENT WITH NUCLEAR CHARGE IZ0
C          (CHARACTER*12 FUNCTION VERSION OF 'XXELEM')
C
C CALLING PROGRAM: GENERAL USE
C
C FUNCTION:
C
C          (C*12) XFELEM  = FUNCTION NAME -
C                           NAME OF ELEMENT WITH NUCLEAR CHARGE 'IZ0'
C          (I*4)  IZ0     = ELEMENT NUCLEAR CHARGE
C
C          (C*12) NAMES() = NAMES OF FIRST 50 ELEMENTS.
C                           ARRAY DIMENSION => NUCLEAR CHARGE
C
C NOTES:    IF NUCLEAR CHARGE IS OUT OF RANGE, I.E.NOT BETWEEN 1 & 50,
C           THEN THE CHARACTER STRING 'XFELEM' IS RETURNED BLANK.
C
C ROUTINES: NONE
C
C
C AUTHOR:   PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C           K1/0/81
C           JET EXT. 4569
C
C DATE:     13/02/91
C
C VERSION: 1.2
C UPDATE:   17/09/99  HUGH SUMMERS - INCREASED ELEMENT NUMBER TO 92
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      INTEGER             IZ0
```

9.75 xfesym: Subroutine xfesym from library adaslib

```
      FUNCTION XFESYM ( IZ0 )
C-----------------------------------------------------------------------
C
C ************** FORTRAN77 CHARACTER*2 FUNCTION: XFESYM ****************
C
C PURPOSE: TO RETURN THE SYMBOL FOR THE ELEMENT WITH NUCLEAR CHARGE IZ0
C          (CHARACTER*2 FUNCTION VERSION OF 'XXESYM')
C
C CALLING PROGRAM: GENERAL USE
C
C FUNCTION:
C
C          (C*2)  XFESYM  = FUNCTION NAME -
C                           SYMBOL OF ELEMENT WITH NUCLEAR CHARGE 'IZ0'
C          (I*4)  IZ0     = ELEMENT NUCLEAR CHARGE
C
C          (C*2)  SYMBOL()= SYMBOLS OF FIRST 92 ELEMENTS.
C                           ARRAY DIMENSION => NUCLEAR CHARGE
C
C NOTES:    IF NUCLEAR CHARGE IS OUT OF RANGE, I.E.NOT BETWEEN 1 & 92,
C           THEN THE CHARACTER STRING 'XFESYM' IS RETURNED BLANK.
C
C ROUTINES: NONE
C
C
C AUTHOR:   PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C           K1/0/81
C           JET EXT. 4569
C
C DATE:     12/02/91
C
C UPDATES:  25/10/94 L. JALOTA (TESSELLA SUPPORT SERVICES PLC)
C      CHANGED CASE OF SYMBOL TO LOWER CASE FOR UNIX
C
C VERSION: 1.2
C UPDATES:  17/09/99 HUGH SUMMERS - INCREASED ELEMENT NUMBER TO 92
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      INTEGER             IZ0
```

9.76 xx0000: Subroutine xx0000 from library adaslib

```
      SUBROUTINE XX0000
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XX0000 *********************
C
C  PURPOSE: ADAS CONFIGURATION FILE FOR SETTING MACHINE DEPENDANT
C           VARIABLES ETC.
C
C  CALLING PROGRAM: ANY MAIN ADAS PROGRAM
C
C
C  SUBROUTINE:
C
C          (I*4)  I4UNIT  = FUNCTION (SEE ROUTINE SECTION BELOW)
C          (I*4)  IDUMP   = GENERAL USE
C
C NOTE:
C
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4UNIT     ADAS      SET UNIT NUMBER FOR OUTPUT OF MESSAGES
C
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/37
C          JET EXT. 5023
C
C DATE:    23/04/93
C
C MODIFIED:  6/3/94 L. JALOTA   CHANGED OUTPUT UNIT TO 0 FOR
C ERROR MESSAGES SO AS NOT TO
C INTERFERE WITH POIPE COMMUN-
C ICATIONS WITH IDL.
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
```

9.77 xxadas: Subroutine xxadas from library adaslib

```
      SUBROUTINE XXADAS( HEADER )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXADAS *********************
C
C  PURPOSE: ADAS ROUTINE   -  GETS AN 80 BYTE CHARACTER STRING HEADER
C           CONTAINING THE ADAS RELEASE & VERSION, THE EXECUTING PROGRAM
C           NAME & VERSION, AND THE CURRENT DATE & TIME FROM IDL VIA
C           THE PIPE.
C
C  CALLING PROGRAM: GENERAL USE
C
C  SUBROUTINE:
C
C  OUTPUT: (C*80) HEADER   = ADAS HEADER (AS DESCRIBED ABOVE)
C
C          (I*4)   PIPEIN   = PARAMETER = UNIT NUMBER FOR OUTPUT TO PIPE
C          (I*4)   PIPEOU   = PARAMETER = UNIT NUMBER FOR INPUT TO PIPE
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C
C AUTHOR:  ANDREW BOWEN (TESSELLA SUPPORT SERVICES PLC)
C
C DATE:    24/05/93
C
C-----------------------------------------------------------------------
      CHARACTER*80        HEADER
```

9.78 xxbasa: Subroutine xxbasa from library adaslib

```
      subroutine xxbasa(a , ia , b , ib , base , c  , ic)
c-----------------------------------------------------------------------
c
c  ****************** fortran77 subroutine: xxbasa *********************
c
c  purpose : to add two numbers, given in the form a*baseˆia and
c            b*baseˆib, where the base must be the same.
c
c  calling program: general use
c
c  subroutine:
c
c  input : (r*8)   a        = mantissa of number
c  input : (i*4)   ia       = characteristic of number
c  input : (r*8)   b        = mantissa of number
c  input : (i*4)   ib       = characteristic of number
c  input : (r*8)   base     = base of number system
c
c  output: (r*8)   c        = mantissa of answer
c  output: (i*4)   ic       = characteristic of answer
c
c routines:
c          routine    source    brief description
c          ------------------------------------------------------------
c          xxbass     adas      to scale a number to a base
c          i4unit     adas      fetch unit number for output of messages
c
c author:  Hugh Summers
c          JET K1/1/61
c          Tel. 01235-46-4459
c date:    14-05-2008
c
c
c version  : 1.1
c date     : 14-05-2008
c modified : Hugh Summers
c              - first version.
c-----------------------------------------------------------------------
      INTEGER             IA,          IB,          IC
      REAL*8              A,           B,           BASE,        C
```

9.79 xxbase: Subroutine xxbase from library adaslib

```
      subroutine xxbase( a , ia , base_a , value , code )
c-----------------------------------------------------------------------
c
c  ****************** fortran77 subroutine: xxbase ********************
c
c  purpose: to produce a number from representation as a*baseˆia
c           protected against underflow and overflow.
c
c  input : (r*8)  a       = mantissa part of number
c  input : (r*8)  ia      = characteristic part of number
c  input : (c*(*))code    = 'zero' => zero a delivered if underflow
c                           'min'  => minimum number delivered
c  output: (r*8)  value   = evaluated number
c
c
c routines:
c          routine    source    brief description
c          ------------------------------------------------------------
c          xxbasr     adas      convert scaled number to new base
c          i4unit     adas      fetch unit number for output of messages
c
c author:  Hugh Summers
c          JET K1/1/61
c          Tel. 01235-46-4459
c date:    07-05-2008
c
c
c version  : 1.1
c date     : 07-05-2008
c modified : Hugh Summers
c              - first version.
c-----------------------------------------------------------------------
      CHARACTER*(*)       CODE
      INTEGER             IA
      REAL*8              A,           BASE_A,      VALUE
```

9.80 xxbasr: Subroutine xxbasr from library adaslib

```
      subroutine xxbasr(a , ia , base_a, b, ib , base_b)
c-----------------------------------------------------------------------
c
c  ****************** fortran77 subroutine: xxbasr *********************
c
c  purpose : to rebase a number, given in the form a*base_a^ia to the
c            form b*base_b^ib.
c
c  calling program: general use
c
c  subroutine:
c
c  input : (r*8)   a        = mantissa of number
c  input : (i*4)   ia       = characteristic of number
c  input : (r*8)   base_a   = base of number system
c  input : (r*8)   base_b   = base of new number system
c
c  output: (r*8)   b        = mantissa of number in new system
c  output: (i*4)   ib       = characteristic of number in new system
c
c routines:
c          routine    source    brief description
c          ------------------------------------------------------------
c          xxbass     adas      to scale a number to a base
c          i4unit     adas      fetch unit number for output of messages
c
c author:  Hugh Summers
c          JET K1/1/61
c          Tel. 01235-46-4459
c date:    07-05-2008
c
c
c version  : 1.1
c date     : 07-05-2008
c modified : Hugh Summers
c              - first version.
c-----------------------------------------------------------------------
      INTEGER             IA,          IB
      REAL*8              A,           B,           BASE_A,      BASE_B
```

9.81 xxbass: Subroutine xxbass from library adaslib

```
      subroutine xxbass(a , ia , b, ib , base)
c-----------------------------------------------------------------------
c
c  ****************** fortran77 subroutine: xxbass *********************
c
c  purpose : to scale a number, given in the form a*baseˆia such that
c            base <= a < 1/base.
c
c  calling program: general use
c
c  subroutine:
c
c  input : (r*8)   a        = mantissa of number
c  input : (i*4)   ia       = characteristic
c  input : (r*8)   base     = base of number system
c
c  output: (r*8)   b        = modified mantissa of number
c  output: (i*4)   ib       = modified characteristic of number
c
c routines:
c          routine    source    brief description
c          ------------------------------------------------------------
c          i4unit     adas      fetch unit number for output of messages
c
c author:  Hugh Summers
c          JET K1/1/61
c          Tel. 01235-46-4459
c date:    07-05-2008
c
c
c version  : 1.1
c date     : 07-05-2008
c modified : Hugh Summers
c              - first version.
c-----------------------------------------------------------------------
      INTEGER             IA,          IB
      REAL*8              A,           B,           BASE
```

9.82 xxbcon: Subroutine xxbcon from library adaslib

```
      SUBROUTINE XXBCON( INTYP, OUTTYP, IEVAL, EIN, EOUT )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXBCON *********************
C
C  PURPOSE: TO CONVERT AN ARRAY OF BEAM ENERGIES INTO SPECIFIED UNITS
C
C  CALLING PROGRAM: GENERAL USE
C
C  SUBROUTINE:
C
C  INPUT :    (I*4)  INTYP   = 1 => 'EIN(ARRAY)' UNITS: EV/AMU
C                            = 2 => 'EIN(ARRAY)' UNITS: AT. UNITS
C                            = 3 => 'EIN(ARRAY)' UNITS: CM SEC-1
C  INPUT :    (I*4)  OUTTYP  = 1 => 'EOUT(ARRAY)' UNITS: EV/AMU
C                            = 2 => 'EOUT(ARRAY)' UNITS: AT. UNITS
C                            = 3 => 'EOUT(ARRAY)' UNITS: CM SEC-1
C  INPUT :    (I*4)  IEVAL   = NUMBER OF BEAM ENERGIES IN 'EIN(ARRAY)'
C  INPUT :    (R*8)  EIN()   = INPUT BEAM ENERGIES (STATED UNITS)
C  OUTPUT:    (R*8)  EOUT()  = OUTPUT BEAM ENERGIES (STATED UNITS)
C
C             (R*8)  EVUATU  = EV/AMU TO ATOMIC UNITS CONVERSION
C             (R*8)  EVUCMS  = EV/AMU TO CM S-1 CONVERSION
C
C             (I*4)  I       = GENERAL USE
C
C             (R*8)  BCONV() = BEAM ENERGY CONVERSION PARAMETERS
C
C ROUTINES:  NONE
C
C NOTE:
C            BEAM ENERGY CONVERSION PARAMETERS:
C
C            INTYP = 1 ; BCONV(1) =             EV/AMU  -> OUTPUT UNITS
C            INTYP = 2 ; BCONV(2) =           AT. UNITS -> OUTPUT UNITS
C            INTYP = 3 ; BCONV(3) =           CM SEC-1  -> OUTPUT UNITS
C
C AUTHOR:  H. P. SUMMERS, UNIVERSITY OF STRATHCLYDE
C          JA8.08
C          TEL. 0141-553-4196
C
C DATE:    20/04/95
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IEVAL,       INTYP,       OUTTYP
      REAL*8              EIN(IEVAL),  EOUT(IEVAL)
```

9.83 xxcase: Subroutine xxcase from library adaslib

```
      subroutine xxcase(input,output,type)


C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXCASE *********************
C
C  PURPOSE: Change a string of arbitrary size into all upper case
C           or all lower case
C
C  CALLING PROGRAM: GENERAL USE.
C
C  INPUT    : (C*(*)) INPUT = Input String
C  INPUT    : (C*2)   TYPE = Type of case to convert to:
C                       'UC' -> Convert to Upper Case
C                       'LC' -> Convert to Lower Case
C                       Anything else -> No conversion
C
C  OUTPUT   : (C*(*)) OUTPUT = Output string in selected case
C
C  ROUTINES : NONE
C
C  AUTHOR   : Allan Whiteford,
C             University of Strathclyde
C
C  VERSION  : 1.1
C  DATE     : 05/09/2001
C  MODIFIED : Allan Whiteford
C             First version.
C
C  VERSION  : 1.2
C  DATE     : 05/05/2005
C  MODIFIED : Martin O'Mullane
C             The routine converted length-1 rather than the whole
C             input string.
C
C-----------------------------------------------------------------------
      CHARACTER*(*)       INPUT,       OUTPUT
      CHARACTER*2         TYPE
```

9.84 xxceia: Subroutine xxceia from library adaslib

```
      SUBROUTINE XXCEIA( EIA )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXCEIA *********************
C
C  PURPOSE: CONVERTS IONISATION POTENTIALS FROM WAVE NUMBERS TO
C           RYDBERGS AND FILLS IN ANY MISSING VALUES UP TO AN ION CHARGE
C           OF 50.
C
C           (IF 'EIA()' IS ALL ZERO - RETURN)
C           (PRIOR TO 16/08/90 WAS KNOWN AS 'E3EIA' - SLIGHTLY AMENDED)
C
C  CALLING PROGRAMS: GENERAL USE
C
C  SUBROUTINE:
C
C  I/O   : (R*8)  EIA()   = IONISATION POTENTIALS: ()=ION CHARGE
C  INPUT :                  UNITS - WAVE NUMBERS (CM-1)
C  OUTPUT:                  UNITS - RYDBERGS
C
C           (R*8) WN2RYD   = WAVE NUMBER (CM-1) TO RYDBERG CONVERSION
C
C           (I*4)  I4UNIT  = FUNCTION (SEE ROUTINE SECTION BELOW)
C           (I*4)  IPOT    = NUMBER OF IONISATION POTENTIAL VALUES
C                            PRESENT IN THE INPUT FILE.
C           (I*4)  I       = GENERAL USE
C
C           (R*8)  A1      = EXTRAPOLATION EQUATION COEFFICIENT
C           (R*8)  A2      = EXTRAPOLATION EQUATION COEFFICIENT
C           (R*8)  A3      = EXTRAPOLATION EQUATION COEFFICIENT
C           (R*8)  XI      = VALUE AT WHICH EXTRAPOLATION IS REQUIRED
C
C ROUTINES:
C           ROUTINE    SOURCE   BRIEF DESCRIPTION
C           ------------------------------------------------------------
C           I4UNIT     ADAS     INTEGER*4 FUNCTION -
C                               FETCH UNIT NUMBER FOR OUTPUT OF MESSAGES
C
C NOTE:
C            THE NUMBER OF IONISATION COEFFICIENTS PRESENT WILL BE EITHER
C            30 OR 50. IF 30 THEN THE VALUES FROM  31  TO  50 NEED TO  BE
C            EXTRAPOLATED. THE EXTRAPOLATION EQUATION  IS  BASED  ON  THE
C            VALUES OF EIA(20), EIA(25) AND EIA(50).
C
C            THE EXTRAPOLATION EQUATION BEING:
C
C            EIA(XI) = A1 + A2*XI + A2*XI*XI
C
C ROUTINES: NONE
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 4569
C
C DATE:    16/08/90
C
C UPDATE:  11/08/93  HP SUMMERS - DO NOT STOP IF IPOT.LE.30, BUT DO NOT
C                                 EXTRAPOLATE
C
```

```
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 06-09-95
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST RELEASE (NO CHANGES)
C
C VERSION: 1.2                          DATE: 20-07-07
C MODIFIED: Allan Whiteford
C               - Small modifications to comments to allow for
C                 automatic documentation preparation.
C
C-------------------------------------------------------------------------
C-------------------------------------------------------------------------
      REAL*8             EIA(50)
```

9.85 xxcftr: Subroutine xxcftr from library adaslib

```
      SUBROUTINE XXCFTR( ICFSEL , CSTRGI , CSTRGO )
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXCFTR *********************
C
C  PURPOSE: CONVERTS A CONFIGURATION CHARACTER STRING, SUCH AS OCCURS
C           IN A SPECIFIC ION FILE LEVEL LIST, BETWEEN EISSNER AND
C           STANDARD FORMS
C
C  CALLING PROGRAMS: GENERAL USE
C
C  SUBROUTINE:
C
C  INPUT : (I*4)   ICFSEL  = 1 => STANDARD FORM OUT, STANDARD FORM IN
C                            2 => EISSNER  FORM OUT, STANDARD FORM IN
C                            3 => STANDARD FORM OUT, EISSNER  FORM IN
C                            4 => EISSNER  FORM OUT, EISSNER  FORM IN
C  INPUT : (C*(*)) CSTRGI  = CONFIGURATION STRING IN INPUT FORM
C  OUTPUT: (C*(*)) CSTRGO  = CONFIGURATION STRING IN OUTPUT FORM
C
C          (I*4)   I       = GENERAL USE
C          (I*4)   ISHEL   = SHELL COUNTER
C          (I*4)   IP      = PARITY OF CONFIGURATION
C          (I*4)   MAXN    = N_SHELL SUM FOR CONFIGURATION
C          (I*4)   NSHEL   = NUMBER OF SHELLS IDENTIFIED fFROM STRING
C          (I*4)   NELA()  = NUMBER OF ELECTRONS IN EACH SHELL
C
C          (C*19)  STRG    = STANDARD FORM CONFIGURATION STRING
C          (C*19)  STRGE   = EISSNER FORM CONFIGURATION STRING
C          (C*1)   CHEISA()= EISSNER CHARACTER FOR ORBITALS
C          (C*2)   CHSTDA()= STANDARD ORBITAL SPEC. FOR EACH SHELL
C                            (EISSNER FORM CASE)
C          (C*1)   CHQA()  = INDEX TO HEXADECIMAL CONVERSIONS
C          (C*1)   CHRA()  = CHAR. FOR NO. OF. EQUIV. ELEC. IN SHELL
C                            (STANDARD FORM CASE)
C
C          (L*4)   LEISS   = .TRUE.  => EISSNER FORM
C                            .FALSE. => NOT EISSNER FORM
C
C
C ROUTINES:
C          ROUTINE    SOURCE   BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4UNIT     ADAS     FETCH UNIT NUMBER FOR OUTPUT OF MESSAGES
C          I4NGRP     ADAS     RETURNS N QUANTUM NUMBER IN THE
C                              EISSNER SINGLE HEXADECIMAL CHARACTER FORM
C          I4PGRP     ADAS     RETURNS PARITY OF ORBITAL GIVEN THE
C                              EISSNER SINGLE HEXADECIMAL CHARACTER FORM
C          I4SCHR     ADAS     RETURNS NUMERICAL VALUE FOR NUMBER OF
C                              EQUIVALENT ELECTRONS GIVEN AS HEX> CHAR.
C          CSTGRP     ADAS     RETURNS TERM OF ORBITAL GIVEN IN THE
C                              EISSNER SINGLE HEXADECIMAL CHARACTER FORM
C          CEIGRP     ADAS     RETURNS EISSNER CODE FOR ORBITAL
C
C
C NOTE:    THE ROUTINE IS USED TO CONVERT THE CONFIGURATION CHARACTER
C          STRING OCCURRING IN ADF04 FILE LEVEL LISTS.  THE STRING
C          LENGTH ALLOCATED TO THIS IS *18 FOLLOWING 1 BLANK SPACE
```

```
C          AFTER THE LEVEL INDEX.  A PROBLEM ARISES WHEN THE FIRST
C          SHELL CONTAINS MORE THAN 9 EQUIVALENT ELECTRONS.  IN THIS
C          CASE, OVERSPILL IS ALLOWED INTO THE BLANK CHARACTER SPACE.
C          THE ROUTINE WILL ANALYSE A *19 STRING INCLUDING THE USUALLY
C          BLANK LOCATION OR A *18 STRING EXCLUDING IT.  IN THE LATTER
C          CASE AN INTELLIGENT GUESS IS MADE AS TO WHETHER THE OMITTED
C          BLANK SHOULD IN FACT BE A '1'.  THIS SITUATION OCCURS FOR A
C          LEADING CLOSED D-SHELL.
C
C AUTHOR:  H. P. SUMMERS, UNIVERSITY OF STRATHCLYDE
C          JA8.08
C          TEL. 0141-553-4196
C
C DATE:    25/10/95
C
C
C UPDATE:  19/02/03  H. P. SUMMERS - EXTENDED RANGE AND STRINGS
C
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 19-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - PUT UNDER SCCS CONTROL
C
C VERSION: 1.2                          DATE: 19-02-03
C MODIFIED: H. P. SUMMERS
C - EXTENDED RANGE AND STRINGS
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      CHARACTER*(*)       CSTRGI,      CSTRGO
      INTEGER             ICFSEL
```

9.86 xxcheb: Subroutine xxcheb from library adaslib

```
      SUBROUTINE XXCHEB( X , Y , N , A , M1 , REF )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXCHEB *********************
C
C  PURPOSE:           CARRY OUT CHEBYSCHEV POLYNOMIAL FIT ALGORITHM
C                     (DIRECT REPLACEMENT FOR NAG MINIMAX POLYNIMIAL
C                      COEFT. ROUTINE E02ACF - has same argument list).
C
C  REFERENCE:         TRANSLATION FROM ALGOL CHEBYSCHEV POLYNOMIAL FIT
C                     ALGORITHM BY -
C                     Boothroyd,
C                     Communications of the ACM, 10(12), December 1967
C
C  CALLING PROGRAMS:  XXMNMX
C
C  SUBROUTINE:
C
C  INPUT : (R*8)  X()     = Array of Input X Co-ordinates
C                           Dimension = N
C  INPUT : (R*8)  Y()     = Array of Input Y Co-ordinates
C                           Dimension = N
C  INPUT : (I*4)  N       = Number of Data Points
C                           Dimension = N
C  OUTPUT: (R*8)  A()     = Coefficients of the Fitted Polynomial.
C                           Dimension = M1
C  INPUT : (I*4)  M1      = M + 1 = The order of the polynomial to be
C                           found + 1. The highest order term is
C                           A(M1)*X(M) !!!
C  OUTPUT: (R*8)  REF     = Final Reference Deviation.
C
C  NOTES:     Based on Revision 1.2 (13:26:01 13OCT94) of XXCHEB by CJW
C             from the University of Strathclyde with the following
C             bug correction -
C               The line (within the DO 100 loop):
C                  IF (I.NE.J)
C               was corrected to:
C                  IF (I.NE.RJ)
C
C  ROUTINES:  None
C
C  AUTHOR:    CJW (University of Strahclyde)
C  EDITED BY: PAUL BRIDEN (Tessella Support Services plc)
C             K1/0/37
C             JET ext. 5023
C
C  DATE:      31/10/94
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             M1,          N
      REAL*8              A(M1),       REF,         X(N),        Y(N)
```

9.87 xxchss: Subroutine xxchss from library adaslib

```
      subroutine xxchss(n, lup, ca, itag)

C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXCHSS ********************
C
C  PURPOSE: Sorts a character array xa and its index array. This is a
C           bubble sort designed for small arrays.
C
C
C  CALLING PROGRAM:  General use
C
C
C  SUBROUTINE:
C
C  INPUT       :  (I*4)  N     = Size of input arrays
C  INPUT       :  (L*4)  LUP   = .TRUE. sort in ascending order
C                                .FALSE. descending order
C  INPUT/OUPUT :  (C)    CA    = Array to be sorted
C  INPUT/OUPUT :  (I*4)  ITAG  = Original index of sorted XA
C
C  ROUTINES    : NONE
C
C  NOTES       : Shell sort from Numerical Receipies.
C
C  AUTHOR      : Martin O'Mullane
C
C  DATE        : 7-02-2000
C
C  VERSION     : 1.1
C
C
C-----------------------------------------------------------------------
      CHARACTER*(*)       CA(*)
      INTEGER             ITAG(*),     N
      LOGICAL             LUP
```

9.88 xxcmps: Subroutine xxcmps from library adaslib

```
      SUBROUTINE XXCMPS( CSTA19 , CSTB19  , LEQUIV  )
C
C----------------------------------------------------------------------
C
C  ***************** FORTRAN77 SUBROUTINE: XXCMPS *********************
C
C  PURPOSE: COMPARES TWO NINETEEN CHARACTER CONFIGURATION STRINGS IN
C           STANDARD FORM AND DETECTS IF THEY ARE EQUIVALENT INDEPENDENT
C           OF THE ORBITALS BEING IN UPPER OR LOWER CASE.
C
C  CALLING PROGRAMS: GENERAL USE
C
C  SUBROUTINE:
C
C  INPUT : (C*19)  CSTA19  = FIRST CONFIGURATION CHARACTER STRING
C  INPUT : (C*19)  CSTB19  = SECOND CONFIGURATION CHARACTER STRING
C  OUTPUT: (L*4)   LEQUIV  = .TRUE.  => CONFIGURATION STRINGS EQUIVALENT
C                            .FALSE. => STRINGS NOT EQUIVALENT
C
C ROUTINES:
C          ROUTINE    SOURCE   BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          XXWORD     ADAS     FINDS NUMBER OF WORDS IN A STRING
C
C AUTHOR:  H. P. SUMMERS, UNIVERSITY OF STRATHCLYDE
C          JA8.08
C          TEL. 0141-553-4196
C
C DATE:    01/10/96
C
C
C UPDATE:
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 14-10-96
C MODIFIED: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C               - PUT UNDER SOFTWARE CONTROL
C
C VERSION: 1.2                          DATE: 28-08-97
C MODIFIED: HUGH SUMMERS
C               - EXTENDED TO INCLUDE 'G' STATES
C
C VERSION: 1.3                          DATE: 26-06-98
C MODIFIED: RICHARD MARTIN
C     - EDITED TO MAKE SURE 'IF' STATEMENT DOESN'T SPAN > 20
C LINES OF TEXT -CRASHES ON SUNOS 4.1.x
C
C----------------------------------------------------------------------
C----------------------------------------------------------------------
      CHARACTER*19        CSTA19,      CSTB19
      LOGICAL             LEQUIV
```

9.89 xxcomm: Subroutine xxcomm from library adaslib

```
      subroutine xxcomm(iunit, ndcomm, comments, ncomm)

C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXCOMM *********************
C
C  PURPOSE: Read the comments from an ADAS adf dataset.
C
C  CALLING PROGRAM: General use.
C
C  INPUT:  (C*(*)) dsname    = dataset name
C  INPUT:  (I*4)   ndcomm    = maximum number of comment lines
C
C  OUTPUT: (I*4)   ncomm     = nunber of comment lines
C  OUTPUT: (C*(*)) comments  = array of comment strings
C
C  ROUTINES : NONE
C
C  AUTHOR   : Martin O'Mullane,
C
C  VERSION  : 1.1
C  DATE     : 14-04-2005
C  MODIFIED : Martin O'Mullane
C              - First version.
C
C-----------------------------------------------------------------------
      CHARACTER*(*)       COMMENTS(NDCOMM)
      INTEGER             IUNIT,       NCOMM,       NDCOMM
```

9.90 xxdams: Subroutine xxdams from library adaslib

```
      SUBROUTINE XXDAMS( VALUE )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXDAMS  *********************
C
C  PURPOSE: ADAS ROUTINE - SETS UP and interrogates a value for
C           the donor  mass.
C
C              VALUE ON INPUT   =>   VALUE ON OUTPUT
C
C                     ?                 returns value
C                  <BLANK>              returns value
C                     *                 returns 'UNSET'
C                  <OTHER>              sets value
C
C
C  CALLING PROGRAM: GENERAL USE - designed for use with R8CONST
C
C  SUBROUTINE:
C
C   I/O   : (C*7)  VALUE   = VALUE UNDER WHICH ADAS DATA IS STORED
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C
C NOTE:
C          TO CHECK CURRENT donor MASS CALL XXDAMS WITH
C          ? AS INPUT.
C
C AUTHOR:  Martin O'Mullane
C
C DATE:    01/04/99
C
C UPDATE:
C
C-----------------------------------------------------------------------
      CHARACTER*(*)       VALUE
```

9.91 xxdata_00: Subroutine xxdata_00 from library adaslib

```
       subroutine xxdata_00( iunit  , dsname ,
     &                       izdimd , iodimd , imdimd ,
     &                       esym   , iz0    , bwnoa  , eeva   ,
     &                       iorba  , na     , la     , iqa    ,
     &                       cstr_std       ,
     &                       imeta  , eevma          ,
     &                       iorbma , nma    , lma    , iqma   ,
     &                       cstrm_std      ,
     &                       lexist         , lresol
     &                     )
c
c-----------------------------------------------------------------------
c
c  ****************** fortran77 subroutine: xxdata_00 ******************
c
c  purpose:  to fetch data from an adf00 data set and detect its main
c            characteristics.
c
c                1. element symbol and nuclear charge
c                2. ionisation potentials (cm-1 and eV)
c                3. shell occupancies in the normal collating order
c
c  calling program: various
c
c           ionisation potential: eV
c           configuration:        standard form nlq (incl. integers
c                                 for n>9 and q>9 , lower case
c                                 letter for l and space separators)
c
c  subroutine:
c
c  input : (i*4)   iunit     = unit to which input file is allocated
c  input : (c*(*)) dsname    = name of opened data set on iunit
c  input : (i*4)   izdimd    = maximum nuclear charge
c  input : (i*4)   iodimd    = max. number of orbitals
c  input : (i*4)   imdimd    = max. number of metastables
c
c  output: (c*2)   esym      = element symbol.
c  output: (i*4)   iz0       =         nuclear charge read
c  output: (r*8)   bwnoa()   = ionisation potential (cm-1) of each stage
c                              1st dim: index = nuclear charge +1
c  output: (r*8)   eeva()    = ionisation potential (eV) of each stage
c                              1st dim: index = nuclear charge +1
c  output: (i*4)   iorba()   = number of orbital shells in configuration
c                              1st dim: index = nuclear charge +1
c  output: (i*4)   na(,)     = principal quantum number of shell
c                              1st dim: index = nuclear charge +1
c                              2nd dim: shell index
c  output: (i*4)   la(,)     = orbital ang. momentum qu. no. of shell
c                              1st dim: index = nuclear charge +1
c                              2nd dim: shell index
c  output: (i*4)   iqa(,)    = occupancy. of shell
c                              1st dim: index = nuclear charge +1
c                              2nd dim: shell index
c
c  output: (c*(*)) cstr_std()= configuration string in standard form
c                              1st dim: index = nuclear charge +1
c
c  output: (r*8)   eevma(,)  = excitation energy (eV) of each metastable
```

```
c                               1st dim: index = nuclear charge +1
c                               2nd dim: index = metastable index
c  output: (i*4)   iorbma(,) = number of orbital shells in metas. config.
c                               1st dim: index = nuclear charge +1
c                               2nd dim: index = metastable index
c  output: (i*4)   nma(,,)    = principal quantum number of metas.shell
c                               1st dim: index = nuclear charge +1
c                               2nd dim: shell index
c                               3rd dim: index = metastable index
c  output: (i*4)   lma(,,)    = orbital ang. mom. qu. no. of metas. shell
c                               1st dim: index = nuclear charge +1
c                               2nd dim: shell index
c                               3rd dim: index = metastable index
c  output: (i*4)   iqma(,,)   = occupancy. of metas. shell
c                               1st dim: index = nuclear charge +1
c                               2nd dim: shell index
c                               3rd dim: index = metastable index
c
c  output: (c*(*)) cstrm_std(,)=meta. config. string in standard form
c                               1st dim: index = nuclear charge +1
c                               2nd dim: index = metastable index
c
c  output: (l*4)   lexist  = .true.  => ionisation potential present
c                          = .false. => not present
c  output: (l*4)   lresol  = .true.  => metastable resolved adf00 file
c                          = .false. => not metastable resolved adf00
c
c
c routines:
c          routine    source    brief description
c          ------------------------------------------------------------
c          i4unit     adas      fetch unit number for output of messages
c          i4fctn     adas      converts from char. to integer  variable
c          xxslen     adas      finds string length excluding leading and
c                               trailing blanks
c          xxword     adas      parses a string into separate words
c                               for ' ()<>{}' delimiters
c          xxcase     adas      changes a string to upper or lower case
c          xfesym     adas      obtain element symbol from nuclear charge
c          xfelem     adas      obtain element name from nuclear charge
c          xxterm     adas      terminate program with a message
c
c
c author:  Hugh Summers, University of Strathclyde
c          JA7.08
c          tel. 0141-548-4196
c
c date:    27/04/04
c
c update:  15/12/06  H. P. Summers - extended to handle metastable resolved
c                                    adf00 files
c
c
c version: 1.1                          date: 27-04-04
c modified: H.P. Summers
c               - first version
c
c version: 1.2                          date: 05-01-07
c modified: H. P. Summers
c               - extended to handle metastable resolved
c                                    adf00 files
```

```
c
c
c-----------------------------------------------------------------------
c-----------------------------------------------------------------------
      CHARACTER*(*)       CSTRM_STD(IZDIMD,IMDIMD), CSTR_STD(IZDIMD)
      CHARACTER*(*)       DSNAME
      CHARACTER*2         ESYM
      INTEGER             IMDIMD,      IMETA(IODIMD),            IODIMD
      INTEGER             IORBA(IZDIMD)
      INTEGER             IORBMA(IZDIMD,IMDIMD),    IQA(IZDIMD,IODIMD)
      INTEGER             IQMA(IZDIMD,IODIMD,IMDIMD),            IUNIT
      INTEGER             IZ0,         IZDIMD,      LA(IZDIMD,IODIMD)
      INTEGER             LMA(IZDIMD,IODIMD,IMDIMD)
      INTEGER             NA(IZDIMD,IODIMD)
      INTEGER             NMA(IZDIMD,IODIMD,IMDIMD)
      LOGICAL             LEXIST,      LRESOL
      REAL*8              BWNOA(IZDIMD),            EEVA(IZDIMD)
      REAL*8              EEVMA(IZDIMD,IODIMD)
```

9.92 xxdata_01: Subroutine xxdata_01 from library adaslib

```
       SUBROUTINE xxdata_01( IUNIT  , MXNENG , MXNSHL ,
     &                       SYMBR  , SYMBD  , IZR    , IZD    ,
     &                       INDD   , NENRGY , NMIN   , NMAX   ,
     &                       LPARMS , LSETL  , ENRGYA ,
     &                       ALPHAA , LFORMA , XLCUTA , PL2A   ,
     &                       PL3A   , SIGTA  , SIGNA  , SIGLA
     &                     )


C-----------------------------------------------------------------------
C
C  ***************** FORTRAN77 SUBROUTINE: xxdata_01 ******************
C
C  PURPOSE:  TO FETCH DATA FROM INPUT DATA SET OF TYPE ADF01.
C
C  CALLING PROGRAM: ADAS301/ADAS306/ADAS307/ADAS308/ADAS309
C
C  DATA:
C
C           THE UNITS USED IN THE DATA FILE ARE TAKEN AS FOLLOWS:
C
C           COLLISION ENERGIES  : KEV/AMU
C           ALPHA               :
C           TOTAL XSECTS.       : CM2
C           N-SHELL XSECTS.     : CM2
C           NL-SHELL DATA       : CM2
C
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  IUNIT     = UNIT TO WHICH INPUT FILE IS ALLOCATED.
C  INPUT : (I*4)  MXNENG    = MAXIMUM NO. OF ENERGIES.
C  INPUT : (I*4)  MXNSHL    = MAXIMUM NO. OF N SHELLS.
C
C  OUTPUT: (C*2)  SYMBR     = READ - RECEIVER ION ELEMENT SYMBOL.
C  OUTPUT: (C*2)  SYMBD     = READ - DONOR ION ELMENT SYMBOL.
C  OUTPUT: (I*4)  IZR       = READ - ION CHARGE OF RECEIVER.
C  OUTPUT: (I*4)  IZD       = READ - ION CHARGE OF DONOR.
C  OUTPUT: (I*4)  INDD      = READ - DONOR STATE INDEX.
C  OUTPUT: (I*4)  NENRGY    = NUMBER OF ENERGIES READ.
C  OUTPUT: (I*4)  NMIN      = LOWEST N-SHELL FOR WHICH DATA READ.
C  OUTPUT: (I*4)  NMAX      = HIGHEST N-SHELL FOR WHICH DATA READ.
C  OUTPUT: (L*4)  LPARMS    = FLAGS IF L-SPLITTING PARAMETERS PRESENT.
C                             .TRUE.  => L-SPLITTING PARAMETERS PRESENT.
C                             .FALSE  => L-SPLITTING PARAMETERS ABSENT.
C  OUTPUT: (L*4)  LSETL     = FLAGS IF L-RESOLVED DATA PRESENT.
C                             .TRUE.  => L-RESOLVED DATA PRESENT.
C                             .FALSE  => L-RESOLVED DATA ABSENT.
C  OUTPUT: (R*8)  ENRGYA()  = READ - COLLISION ENERGIES.
C                             UNITS: EV/AMU (READ AS KEV/AMU)
C                             DIMENSION: ENERGY INDEX
C  OUTPUT: (R*8)  ALPHAA()  = READ - EXTRAPOLATION PARAMETER ALPHA.
C                             DIMENSION: ENERGY INDEX
C  OUTPUT: (I*4)  LFORMA()  = READ - PARAMETERS FOR CALCULATING L-RES
C                                    X-SEC.
C                             DIMENSION: ENERGY INDEX
C  OUTPUT: (R*8)  XLCUTA()  = READ - PARAMETERS FOR CALCULATING L-RES
C                                    X-SEC.
C                             DIMENSION: ENERGY INDEX
```

```
C  OUTPUT: (R*8)  PL2A()    = READ - PARAMETERS FOR CALCULATING L-RES
C                                    X-SEC.
C                              DIMENSION: ENERGY INDEX
C  OUTPUT: (R*8)  PL3A()    = READ - PARAMETERS FOR CALCULATING L-RES
C                                    X-SEC.
C                              DIMENSION: ENERGY INDEX
C  OUTPUT: (R*8)  SIGTA()   = READ - TOTAL CHARGE EXCHANGE
C                                    CROSS-SECTION.
C                              UNITS: CM2
C                              DIMENSION: ENERGY INDEX
C  OUTPUT: (R*8)  SIGNA(,)  = READ - N-RESOLVED CHARGE EXCHANGE
C                                    CROSS-SECTIONS.
C                              UNITS: CM2
C                              1ST DIMENSION: ENERGY INDEX
C                              2ND DIMENSION: N-SHELL
C  OUTPUT: (R*8)  SIGLA(,)  = READ - L-RESOLVED CHARGE EXCHANGE
C                                    CROSS-SECTIONS.
C                              UNITS: CM2
C                              1ST DIMENSION: ENERGY INDEX
C                              2ND DIMENSION: INDEXED BY I4IDFL(N,L)
C
C          (R*8)  ZEROST   = PARAMETER = EFFECTIVE SHIFT APPLIED TO
C                                        CROSS-SECTION VALUES TO AVOID
C                                        ZERO VALUES (WILL NOT AFFECT
C                                        ANY VALUES WHICH ARE GREATER
C                                        THAN AROUND 1.0E+15*ZEROSHFT -
C                                        i.e. 1.0E-25.)
C
C          (I*4)  OLDMIN   = PREVIOUS VALUE READ FOR NMIN.
C          (I*4)  OLDMAX   = PREVIOUS VALUE READ FOR NMAX.
C          (I*4)  IBLK     = CURRENT DATA BLOCK.
C          (I*4)  IVALUE   = USED TO PARSE FOR END OF DATA FLAG (-1).
C          (I*4)  N        = N QUANTUM NUMBER.
C          (I*4)  L        = L QUANTUM NUMBER.
C          (I*4)  I        = LOOP COUNTER.
C          (I*4)  J        = LOOP COUNTER.
C          (I*4)  IERR     = ERROR RETURN CODE.
C          (C*2)  CIZR     = ION CHARGE OF RECEIVER.
C          (C*2)  CIZD     = ION CHARGE OF DONOR.
C          (C*1)  INDD     = DONOR STATE INDEX.
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4FCTN     ADAS      RETURNS CHARACTER STRING AS AN INTEGER.
C          I4UNIT     ADAS      FETCH UNIT NUMBER FOR OUTPUT OF MESSAGES
C          I4IDFL     ADAS      RETURNS UNIQUE INDEX FROM QUANTUM
C                               NUMBERS N AND L.
C          XXIDTL     ADAS      INVERSE OF I4IDFL. RETURNS QUANTUM
C                               NUMBERS N AND L FROM INDEX.
C
C AUTHOR:  JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 5183
C
C DATE:    21/09/93
C
C UPDATE:  18/10/93 - J NASH    - ADAS91:
C          UPDATED TO READ L-SPLITTING PARAMETERS IF PRESENT IN DATASET.
C
C UPDATE:  01/05/95 - Tim Hammond - IDLADAS:
```

```
C          UNIX port.
C
C UPDATE:  16/05/95 - Tim Hammond - IDLADAS:
C          ADDED AND APPLIED ZEROST PARAMETER => EFFECTIVE ZERO FOR
C          CROSS-SECTIONS (CODING DONE BY PAUL BRIDEN).
C
C-----------------------------------------------------------------------
C
C
C NOTES: Copied from cxdata.for.
C        Remove the redundant titled from argument list.
C        This is v1.1 of xxdata_01.
C
C
C VERSION  : 1.1
C DATE     : 14-09-2004
C MODIFIED : Martin O'Mullane
C              - First version.
C
C VERSION  : 1.2
C DATE     : 26-04-2007
C MODIFIED : Hugh Summers
C             - Remove unused m-subshell data possibility.
C
C VERSION  : 1.3
C DATE     : 22-05-2007
C MODIFIED : Martin O'Mullane
C              - Initialize output arrays to zero.
C
C VERSION  : 1.4
C DATE     : 12-06-2008
C MODIFIED : Allan Whiteford
C              - Correctly parse files which contain m-subshell
C                data and print a warning to say that the m-resolved
C                data were ignored.
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      CHARACTER*2         SYMBD,       SYMBR
      INTEGER             INDD,        IUNIT,       IZD,         IZR
      INTEGER             LFORMA(MXNENG),           MXNENG,      MXNSHL
      INTEGER             NENRGY,      NMAX,        NMIN
      LOGICAL             LPARMS,      LSETL
      REAL*8              ALPHAA(MXNENG),           ENRGYA(MXNENG)
      REAL*8              PL2A(MXNENG),             PL3A(MXNENG)
      REAL*8              SIGLA(MXNENG,(MXNSHL*(MXNSHL+1))/2)
      REAL*8              SIGNA(MXNENG,MXNSHL),     SIGTA(MXNENG)
      REAL*8              XLCUTA(MXNENG)
```

9.93 xxdata_02: Subroutine xxdata_02 from library adaslib

```
       SUBROUTINE xxdata_02( IUNIT  , DSNAME ,
     &                       NSTORE , NEDIM  ,
     &                       NBSEL  , ISELA  ,
     &                       CPRIMY , CSECDY , CTYPE,
     &                       AMPA   , AMSA   , ALPHA  , ETHRA  ,
     &                       IEA    ,
     &                       TEEA   , SIA
     &                     )
C-----------------------------------------------------------------------
C
C  ***************** FORTRAN77 SUBROUTINE: xxdata_02 ******************
C
C  PURPOSE:  TO  FETCH  DATA  FROM  INPUT ION/ATOM CROSS-SECTION
C            FILES OF TYPE ADF02.
C
C  CALLING PROGRAM: ADAS302/SSIA
C
C  DATA:
C
C           UP TO 'NSTORE' SETS (DATA-BLOCKS) OF DATA MAY BE  READ FROM
C           THE FILE - EACH BLOCK FORMING A COMPLETE SET OF  CROSS-
C           SECTION VALUES FOR GIVEN COLLISION ENERGIES.
C           EACH DATA-BLOCK  IS  ANALYSED INDEPENDENTLY OF ANY  OTHER
C           DATA-BLOCK.
C
C           THE UNITS USED IN THE DATA FILE ARE TAKEN AS FOLLOWS:
C
C           COLLISION ENERGIES  : EV/AMU
C           CROSS-SECTION       : CM**2
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  IUNIT    = UNIT TO WHICH INPUT FILE IS ALLOCATED.
C  INPUT : (C*80) DSNAME   = MVS DATA SET NAME OF DATA SET BEING READ
C
C  INPUT : (I*4)  NSTORE   = MAXIMUM NUMBER  OF  INPUT DATA-BLOCKS  THAT
C                            CAN BE STORED.
C  INPUT : (I*4)  NEDIM    = MAX NUMBER OF COLLISION ENERGIES ALLOWED
C
C  OUTPUT: (I*4)  NBSEL    = NUMBER OF DATA-BLOCKS ACCEPTED & READ IN.
C  OUTPUT: (I*4)  ISELA()  = READ - DATA-SET DATA-BLOCK ENTRY INDICES
C                            DIMENSION: DATA-BLOCK INDEX
C
C  OUTPUT: (C*5)  CPRIMY() = READ - PRIMARY SPECIES IDENTIFICATION
C                            DIMENSION: DATA-BLOCK INDEX
C  OUTPUT: (C*5)  CSECDY() = READ - SECONDARY SPECIES IDENTIFICATION
C                            DIMENSION: DATA-BLOCK INDEX
C  OUTPUT: (C*3)  CTYPE()  = READ - CROSS-SECTION TYPE
C                            DIMENSION: DATA-BLOCK INDEX
C
C  OUTPUT: (R*8)  AMPA()   = READ - PRIMARY SPECIES ATOMIC MASS NUMBER
C                            DIMENSION: DATA-BLOCK INDEX
C  OUTPUT: (R*8)  AMSA()   = READ - SECONDARY SPECIES ATOMIC MASS NUMBER
C                            DIMENSION: DATA-BLOCK INDEX
C  OUTPUT: (R*8)  ALPHA()  = READ - HIGH ENERGY EXTRAPOLATION PARM.
C                            DIMENSION: DATA-BLOCK INDEX
C  OUTPUT: (R*8)  ETHRA()  = READ - ENERGY THRESHOLD (EV)
C                            DIMENSION: DATA-BLOCK INDEX
C
```

```
C  OUTPUT: (I*4)  IEA()    = READ - NUMBER OF COLLISION ENERGIES
C                             DIMENSION: DATA-BLOCK INDEX
C
C  OUTPUT: (R*8)  TEEA(,)  = READ - COLLISION ENERGIES (UNITS: eV/AMU)
C                             1st DIMENSION: COLLISION ENERGY INDEX
C                             2nd DIMENSION: DATA-BLOCK INDEX
C
C  OUTPUT: (R*8)  SIA(,)   =READ - FULL SET OF COLLISION CROSS-
C                                    SECTION VALUES (cm**2)
C                             1st DIMENSION: COLLISION ENERGY INDEX
C                             2nd DIMENSION: DATA-BLOCK INDEX
C
C          (I*4)  I4EIZ0   = FUNCTION - (SEE ROUTINES SECTION BELOW)
C          (I*4)  I4FCTN   = FUNCTION - (SEE ROUTINES SECTION BELOW)
C          (I*4)  I4UNIT   = FUNCTION - (SEE ROUTINE SECTION BELOW)
C          (I*4)  IBLK     = ARRAY INDEX: DATA-BLOCK INDEX
C          (I*4)  ITT      = ARRAY INDEX: COLLISION ENERGY INDEX
C          (I*4)  NENUM    = NUMBER OF COLLISION ENERGIES FOR CURRENT
C                             DATA-BLOCK
C          (I*4)  IABT     = RETURN CODE FROM 'I4FCTN'
C          (I*4)  IPOS2    = GENERAL USE STRING INDEX VARIABLE
C
C          (R*8)  R8FCTN   = FUNCTION - (SEE ROUTINES SECTION BELOW)
C
C          (L*4)  LBEND    = IDENTIFIES WHETHER THE LAST OF THE  INPUT
C                             DATA SUB-BLOCKS HAS BEEN LOCATED.
C                             (.TRUE. => END OF SUB-BLOCKS REACHED)
C
C          (C*1)  CSLASH   = '/' - DELIMITER FOR 'XXHKEY'
C          (C*2)  C2       = GENERAL USE TWO BYTE CHARACTER STRING
C          (C*1)  CKEY1    = 'P'      - INPUT BLOCK HEADER KEY
C          (C*1)  CKEY2    = 'S'      - INPUT BLOCK HEADER KEY
C          (C*1)  CKEY3    = 'A'      - INPUT BLOCK HEADER KEY
C          (C*1)  CKEY4    = 'E'      - INPUT BLOCK HEADER KEY
C          (C*4)  CKEY5    = 'T'      - INPUT BLOCK HEADER KEY
C          (C*3)  C3       = GENERAL USE THREE BYTE CHARACTER STRING
C          (C*9)  C10      = GENERAL USE NINE BYTE CHARACTER STRING
C          (C*80) C80      = GENERAL USE 80 BYTE  CHARACTER  STRING  FOR
C                             THE INPUT OF DATA-SET RECORDS.
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          XXHKEY     ADAS      OBTAIN KEY/RESPONSE STRINGS FROM TEXT
C          I4FCTN     ADAS      INTEGER*4 FUNCTION    -
C                               CONVERT CHARACTER STRING TO INTEGER
C          I4UNIT     ADAS      INTEGER*4 FUNCTION    -
C                               FETCH UNIT NUMBER FOR OUTPUT OF MESSAGES
C          R8FCTN     ADAS      REAL*8 FUNCTION       -
C                               CONVERT CHARACTER STRING TO REAL*8
C
C AUTHOR:  H. P. SUMMERS, UNIVERSITY OF STRATHCLYDE
C          JA8.08
C          TEL. 0141-553-4196
C
C DATE:    12/11/96
C
C UNIX-IDL PORT: H.P.SUMMERS
C
C VERSION: 1.1                          DATE: 19-11-96
C MODIFIED: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
```

```
C           - PUT UNDER S.C.C.S. CONTROL
C
C VERSION: 1.2                          DATE: 14-02-97
C MODIFIED: RICHARD MARTIN
C               - CHANGED INITIALISATION 'CKEY2 /'S  '/' TO 'CKEY2 /'S'/ '
C
C
C-----------------------------------------------------------------------
C
C NOTES: Copied from c2data.for.
C        This is v1.1 of xxdata_02.
C
C
C VERSION  : 1.1
C DATE     : 01-12-2005
C MODIFIED : Martin O'Mullane
C              - First version.
C
C-----------------------------------------------------------------------
      CHARACTER*5         CPRIMY(NSTORE),           CSECDY(NSTORE)
      CHARACTER*3         CTYPE(NSTORE)
      CHARACTER*80        DSNAME
      INTEGER             IEA(NSTORE), ISELA(NSTORE),            IUNIT
      INTEGER             NBSEL,       NEDIM,       NSTORE
      REAL*8              ALPHA(NSTORE),            AMPA(NSTORE)
      REAL*8              AMSA(NSTORE),             ETHRA(NSTORE)
      REAL*8              SIA(NEDIM,NSTORE),        TEEA(NEDIM,NSTORE)
```

9.94 xxdata_03: Subroutine xxdata_03 from library adaslib

```
       SUBROUTINE xxdata_03( IUNIT,  IZDIMD, IGDIMD,
     &                       IZ0,    IZL,    IZU,
     &                       IZRA,   IZDA,   IZIA,    IZTA,    IZSA,
     &                       CRRCA,  NRRCA,  ISRRCA,
     &                       NZA,    KSIA,
     &                       N0RA,   V0RA,   PHFCRA,  EDSPRA,  SCLERA,
     &                       CDRCA,  NDRCA,  ISDRCA,
     &                       DEDA,   FDA,    GDA,     NNDA,    MSDA,
     &                       ITYPDA, N0DA,   NCUTA,   V0DA,    PHFCDA,
     &                       CRFCDA, EPSIJA, FIJA,    EDSPDA,  SCLEDA,
     &                       CCIOA,  NCIOSA, NCIORA,  ISCIOA,
     &                       PIOA,   AIOA,   BIOA,    CIOA,    NQIOA,
     &                       ZETAA,  EIONA,  CIA,
     &                       WGHTA,  ENERA,  CRA,
     &                       CPLTA,  NPLTA,  ISPLTA,
     &                       DEPTA,  FPTA,   GPTA,    NNPTA,   SPYLTA,
     &                       CPLSA,  NPLSA,  ISPLSA,  INFO,
     &                       DEPSA,  FPSA,   GPSA,    NNPSA,   SPYLSA,
     &                       LVALID
     &                     )

C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: D8DATA *********************
C
C  PURPOSE:  TO FETCH DATA FROM INPUT ATOMPARS DATA SET OF TYPE ADF03.
C
C  CALLING PROGRAM: ADAS408
C
C
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  IUNIT     = UNIT TO WHICH INPUT FILE IS ALLOCATED
C  INPUT : (I*4)  IZDIMD    = MAXIMUM NUMBER OF IONISATION STAGES
C  INPUT : (I*4)  IGDIMD    = MAXIMUM NUMBER OF GROUPS
C
C
C  THE OUTPUT ARRAYS ARE INDEXED
C            XXX()  = XXX(IZDIMD)         1ST DIMENSION ION STAGE
C
C            XXX(,) = XXX(IZDIMD,IGDIMD)  1ST DIMENSION ION STAGE
C                                         2ND DIMENSION GROUP
C
C
C
C  OUTPUT: (I*4)  IZ0       =  NUCLEAR CHARGE
C  OUTPUT: (I*4)  IZL       =  LOWEST INCLUDED ION
C  OUTPUT: (I*4)  IZU       =  HIGHEST INCLUDED ION
C
C  OUTPUT: (I*4)  IZRA()    =  RECOMBINING ION (RAD.  RECOM.)
C  OUTPUT: (I*4)  IZDA()    =  RECOMBINING ION (DIEL. RECOM.)
C  OUTPUT: (I*4)  IZIA()    =  IONISING ION    (COLL. IONIS.)
C  OUTPUT: (I*4)  IZTA()    =  RADIATING ION   (TOTAL LINE POWER)
C  OUTPUT: (I*4)  IZSA()    =  RADIATING ION   (SPECIFIC LINE POWER)
C
C
C
C  OUTPUT: (C*5)  CRRCA()   =  RADIATIVE RECOM. CODE
```

```
C  OUTPUT: (I*4)  NRRCA()    =    - NOT USED -
C  OUTPUT: (I*4)  ISRRCA()   =    - NOT USED -
C
C  OUTPUT: (I*4)  NZA()      = LOWEST ACCESSIBLE SHELL FOR RAD. RECOM.
C  OUTPUT: (I*4)  KSIA()     = NUMBER OF ELECTRONS IN SHELL
C
C  OUTPUT: (I*4)  N0RA()     = LOWEST ACCESSIBLE PRINC. QUANTUM SHELL
C                              FOR RAD. RECOM.
C  OUTPUT: (I*4)  V0RA()     = EFFECTIVE PRINCIPAL QUANTUM NUMBER
C                              FOR SHELL
C  OUTPUT: (R*8)  PHFCRA()   = PHASE SPACE OCCUPANCY AVAILABILITY
C                              FOR SHELL
C  OUTPUT: (R*8)  EDSPRA()   = ENERGY ADJUSTMENT IN LOWEST SHELL
C                              RATE COEFFICIENT
C  OUTPUT: (R*8)  SCLERA()   = MULTIPLIER FOR LOWEST SHELL
C                              RATE COEFFICIENT
C
C
C
C  OUTPUT: (C*5)  CDRCA()    =  DIELECTRONIC RECOM. CODE
C  OUTPUT: (I*4)  NDRCA()    =  NUMBER OF TRANSITIONS FOLLOWING
C  OUTPUT: (I*4)  ISDRCA()   =    - NOT USED -
C
C  OUTPUT: (R*8)  DEDA(,)    = TRANSITION ENERGY (EV)
C  OUTPUT: (R*8)  FDA(,)     = OSCILLATOR STRENGTH
C  OUTPUT: (R*8)  GDA(,)     = GAUNT FACTOR
C  OUTPUT: (I*4)  NNDA(,)    = DELTA N FOR TRANSITION
C  OUTPUT: (I*4)  MSDA(,)    = MERTZ SWITCH (0=OFF, 1=ON)
C
C  OUTPUT: (I*4)  ITYPDA(,)  = TYPE OF DIELECTRONIC TRANSITION
C  OUTPUT: (I*4)  N0DA(,)    = LOWEST ACCESSIBLE PRINC. QUANTUM SHELL
C                              FOR DIEL. RE
C  OUTPUT: (I*4)  NCUTA(,)   = CUT-OFF PRINC. QUANTUM SHELL IN
C                              GENERAL PROGRAM
C  OUTPUT: (I*4)  V0DA(,)    = EFFECTIVE PRINC. QUANTUM NUMBER
C                              FOR LOWEST ACCESS
C  OUTPUT: (R*8)  PHFCDA(,)  = PHASE SPACE OCCUPANCY AVAILABILITY
C                              FOR LOWEST SHELL
C  OUTPUT: (R*8)  CRFCDA(,)  = ADJUSTMENT FOR BETHE CORRECTIONS
C                              IN GENERAL PROGRAM
C  OUTPUT: (R*8)  EPSIJA(,)  = Z-SCALED PARENT TRANSITION ENERGY (RYD)
C  OUTPUT: (R*8)  FIJA(,)    = OSCILLATOR STRENGTH FOR TRANSITION
C  OUTPUT: (R*8)  EDSPDA(,)  = ENERGY ADJUSTMENT IN BURGESS GENERAL
C                              FORMULA (RYD)
C  OUTPUT: (R*8)  SCLEDA(,)  = MULTIPLIER ON BURGESS GENERAL FORMULA
C
C
C
C  OUTPUT: (C*5)  CCIOA()    =  COLLISIONAL IONIS. CODE
C  OUTPUT: (I*4)  NCIOSA()   =  NUMBER OF SHELL  VALUES FOLLOWING
C  OUTPUT: (I*4)  NCIORA()   =  NUMBER OF RESON. VALUES FOLLOWING
C  OUTPUT: (I*4)  ISCIOA()   =    - NOT USED -
C
C  OUTPUT: (R*8)  PIOA(,)    = SHELL IONISATION POTENTIAL (EV)
C  OUTPUT: (R*8)  AIOA(,)    = LOTZ PARAMETER
C  OUTPUT: (R*8)  BIOA(,)    = LOTZ PARAMETER
C  OUTPUT: (R*8)  CIOA(,)    = LOTZ PARAMETER
C  OUTPUT: (I*4)  NQIOA(,)   = EQUIVALENT ELECTRONS IN SHELL
C
C  OUTPUT: (R*8)  ZETAA(,)   = NUMBER OF EQUIVALENT ELECTRONS FOR SHELL
C  OUTPUT: (R*8)  EIONA(,)   = IONISATION ENERGY FOR SHELL (RYD)
```

```
C  OUTPUT: (R*8)  CIA(,)    = MULTIPLIER FOR BURGESS-CHIDICHIMO RATE
C                             FOR SHELL
C  OUTPUT: (R*8)  WGHTA(,)  = WEIGHTING FACTOR FOR EXCITATION TO
C                             RESONANCE
C  OUTPUT: (R*8)  ENERA(,)  = EXCITATION ENERGY FOR TRANSITION
C                             TO RESONANCE (RYD)
C  OUTPUT: (R*8)  CRA(,)    = MULTIPLIER ON EXCITATION RATE EXPRESSSION
C
C
C
C  OUTPUT: (C*5)  CPLTA()   =  TOTAL LINE POWER CODE
C  OUTPUT: (I*4)  NPLTA()   =  NUMBER OF TRANSITIONS FOLLOWING
C  OUTPUT: (I*4)  ISPLTA()  =    - NOT USED -
C
C  OUTPUT: (R*8)  DEPTA(,)  = TRANSITION ENERGY (EV)
C  OUTPUT: (R*8)  FPTA(,)   = OSCILLATOR STRENGTH
C  OUTPUT: (R*8)  GPTA(,)   = GAUNT FACTOR
C  OUTPUT: (I*4)  NNPTA(,)  = DELTA N FOR TRANSITION
C
C  OUTPUT: (R*8)  SPYLTA(,) = MULTIPLIER OF VAN REGEMORTER P
C                             FACTOR IN TOTAL POWER
C
C
C
C  OUTPUT: (C*5)  CPLSA()   = SPECIFIC LINE POWER CODE
C  OUTPUT: (I*4)  NPLSA()   =   - NOT USED -
C  OUTPUT: (I*4)  ISPLSA()  =   - NOT USED -
C  OUTPUT: (C*8)  INFO()    = WAVELENGTH OF SPECIFIC LINE FOR
C                             NAMING PURPOSES
C
C  OUTPUT: (R*8)  DEPSA(,)  = TRANSITION ENERGY (EV)
C  OUTPUT: (R*8)  FPSA(,)   = OSCILLATOR STRENGTH
C  OUTPUT: (R*8)  GPSA(,)   = GAUNT FACTOR
C  OUTPUT: (I*4)  NNPSA(,)  = DELTA N FOR TRANSITION
C
C  OUTPUT: (R*8)  SPYLSA(,) = MULTIPLIER OF VAN REGEMORTER P FACTOR
C                             IN SPECIFIC LINE POWER
C
C
C
C  OUTPUT: (L*4)  LVALID  = .TRUE. DATA SET READ AND APPEARS VALID
C                         = .FALSE. ERROR DETECTED IN READING DATA SET
C
C
C
C  PROGRAM:
C          (C*50) ERRMSG()= ERROR MESSAGE STRING
C          (I*4)  IDUM    = PROGRAM USE
C
C
C  ROUTINES:
C          ROUTINE   SOURCE   DESCRIPTION
C          ------------------------------------------------------
C          I4UNIT    ADAS     FETCH UNIT NUMBER FOR MESSAGE OUTPUT
C
C
C  AUTHOR:  H. P. SUMMERS, JET
C          K1/1/57
C          JET EXT. 4941
C
C  DATE:    10/05/94
```

```
C
C
C NOTES: Copied from d8data.for
C
C
C VERSION  : 1.1
C DATE     : 19-07-2003
C MODIFIED : Martin O'Mullane
C              - First version
C
C VERSION  : 1.2
C DATE     : 16-01-2004
C MODIFIED : Martin O'Mullane
C              - V0 in type B radiative recombination is real, not
C                an integer.
C
C-----------------------------------------------------------------------
C     GENERAL VARIABLES

      CHARACTER*5         CCIOA(IZDIMD),            CDRCA(IZDIMD)
      CHARACTER*5         CPLSA(IZDIMD),            CPLTA(IZDIMD)
      CHARACTER*5         CRRCA(IZDIMD)
      CHARACTER*8         INFO(IZDIMD)
      INTEGER             IGDIMD,      ISCIOA(IZDIMD)
      INTEGER             ISDRCA(IZDIMD),           ISPLSA(IZDIMD)
      INTEGER             ISPLTA(IZDIMD),           ISRRCA(IZDIMD)
      INTEGER             ITYPDA(IZDIMD,IGDIMD),    IUNIT,       IZ0
      INTEGER             IZDA(IZDIMD),             IZDIMD
      INTEGER             IZIA(IZDIMD),             IZL
      INTEGER             IZRA(IZDIMD),             IZSA(IZDIMD)
      INTEGER             IZTA(IZDIMD),             IZU
      INTEGER             KSIA(IZDIMD),             MSDA(IZDIMD,IGDIMD)
      INTEGER             N0DA(IZDIMD,IGDIMD),      N0RA(IZDIMD)
      INTEGER             NCIORA(IZDIMD),           NCIOSA(IZDIMD)
      INTEGER             NCUTA(IZDIMD,IGDIMD),     NDRCA(IZDIMD)
      INTEGER             NNDA(IZDIMD,IGDIMD),      NNPSA(IZDIMD,IGDIMD)
      INTEGER             NNPTA(IZDIMD,IGDIMD),     NPLSA(IZDIMD)
      INTEGER             NPLTA(IZDIMD),            NQIOA(IZDIMD,IGDIMD)
      INTEGER             NRRCA(IZDIMD),            NZA(IZDIMD)
      LOGICAL             LVALID
      REAL*8              AIOA(IZDIMD,IGDIMD),      BIOA(IZDIMD,IGDIMD)
      REAL*8              CIA(IZDIMD,IGDIMD),       CIOA(IZDIMD,IGDIMD)
      REAL*8              CRA(IZDIMD,IGDIMD)
      REAL*8              CRFCDA(IZDIMD,IGDIMD),    DEDA(IZDIMD,IGDIMD)
      REAL*8              DEPSA(IZDIMD,IGDIMD),     DEPTA(IZDIMD,IGDIMD)
      REAL*8              EDSPDA(IZDIMD,IGDIMD),    EDSPRA(IZDIMD)
      REAL*8              EIONA(IZDIMD,IGDIMD),     ENERA(IZDIMD,IGDIMD)
      REAL*8              EPSIJA(IZDIMD,IGDIMD),    FDA(IZDIMD,IGDIMD)
      REAL*8              FIJA(IZDIMD,IGDIMD),      FPSA(IZDIMD,IGDIMD)
      REAL*8              FPTA(IZDIMD,IGDIMD),      GDA(IZDIMD,IGDIMD)
      REAL*8              GPSA(IZDIMD,IGDIMD),      GPTA(IZDIMD,IGDIMD)
      REAL*8              PHFCDA(IZDIMD,IGDIMD),    PHFCRA(IZDIMD)
      REAL*8              PIOA(IZDIMD,IGDIMD)
      REAL*8              SCLEDA(IZDIMD,IGDIMD),    SCLERA(IZDIMD)
      REAL*8              SPYLSA(IZDIMD,IGDIMD)
      REAL*8              SPYLTA(IZDIMD,IGDIMD),    V0DA(IZDIMD,IGDIMD)
      REAL*8              V0RA(IZDIMD),             WGHTA(IZDIMD,IGDIMD)
      REAL*8              ZETAA(IZDIMD,IGDIMD)
```

9.95 xxdata_04: Subroutine xxdata_04 from library adaslib

```
       subroutine xxdata_04( iunit  ,
     &                       ndlev  , ndtrn , ndmet   , ndqdn , nvmax ,
     &                       titled , iz    , iz0     , iz1   , bwno  ,
     &                       npl    , bwnoa , lbseta  , prtwta, cprta ,
     &                       il     , qdorb , lqdorb  , qdn   , iorb  ,
     &                       ia     , cstrga, isa     , ila   , xja   ,
     &                       wa     ,
     &                       cpla   , npla  , ipla    , zpla  ,
     &                       nv     , scef  ,
     &                       itran  , maxlev,
     &                       tcode  , i1a   , i2a     , aval  , scom  ,
     &                       beth   ,
     &                       iadftyp, lprn  , lcpl    , lorb  , lbeth ,
     &                       letyp  , lptyp , lrtyp   , lhtyp , lityp ,
     &                       lstyp  , lltyp , itieactn, ltied
     &                   )

C-----------------------------------------------------------------------
C
C  ****************** fortran77 subroutine: xxdata_04 ******************
C
C  PURPOSE:  To fetch data from an adf04 data set and detect its main
C            characteristics. This is a fully inclusive version, based
C            on badata.for, detecting the following:
C
C                1. Multiple parent data on the first line including
C                   the j-resolved case
C                2. Supplementary parent assignment data on level
C                   lines for improved automatic ionisation calculation
C                3. Orbital energy data on the level terminator line
C                4. First bethe coefft. at end of e-transition lines for
C                   improved asymptotics
C                5. All transition line qualifiers , 'h',r',s',i','p'
C                   in upper or lower case;  ' ','1','2','3' electron
C                   impact transition types; multiple parents in 'r',
C                   'i',s' transition lines.
C                6. Doubly excited 'r' lines with Auger rate and resonance
C                   capture.
C                7. 'l' lines for dielectronic power correction to singly
C                   excited levels, including effective mean wavelength.
C
C  calling program: various
C
C  data:
C           The 'real' data in the file is represented in an abbreviated
C           form which omits the "d" or "e" exponent specifier.
C           e.g. 1.23d-06 or 1.23e-06 is represented as 1.23-06
C                6.75d+07 or 6.75e+07 is represented as 6.75+07
C
C           Therefore the form of each 'real' number in the data set is:
C                          n.nn+nn or n.nn-nn
C
C           The units used in the data file are taken as follows:
C
C           ionisation potential: wave number (cm-1)
C           index level energies: wave number (cm-1)
C           temperatures        : kelvin
C           a-values            : sec-1
C           gamma-values        :
```

```
C           rate coefft.        : cm3 sec-1
C
C
C  subroutine:
C
C  input : (i*4)  iunit   = unit to which input file is allocated
C  input : (i*4)  ndlev   = maximum number of levels that can be read
C  input : (i*4)  ndtrn   = max. number of transitions that can be read
C  input : (i*4)  nvmax   = max. number of temperatures that can be read in.
C
C  input : (i*4)  itieactn= 1 return data even if some levels are untied.
C                           0 default behaviour - terminate if untied
C                             levels are present.
C                           On output 1 if untied levels present
C                                     0 for no untied levels.
C
C  output: (c*3)  titled  = element symbol.
C  output: (i*4)  iz      =  recombined ion charge read
C  output: (i*4)  iz0     =         nuclear charge read
C  output: (i*4)  iz1     = recombining ion charge read
C                           (note: iz1 should equal iz+1)
C  output: (r*8)  bwno    = ionisation potential (cm-1) of lowest parent
C  output: (i*4)  npl     = number of parents on first line and used
C                           in level assignments
C  output: (r*8)  bwnoa() = ionisation potential (cm-1) of parents
C  output: (l*4)  lbseta()= .true.  - parent weight set for bwnoa()
C                           .false. - parent weight not set for bwnoa()
C  output: (r*8)  prtwta()= parent weight for bwnoa()
C  output: (c*9)  cprta() = parent name in brackets
C
C  output: (i*4)  il      = input data file: number of energy levels
C  output: (r*8)  qdorb() = quantum defects for orbitals
C                           1st dim: index for nl orbital (cf i4idfl.for)
C  output: (l*4)  lqdorb()= .true.  => source data available for qd.
C                         = .false. => source data not availabe qd.=0.0
C  output: (r*8)  qdn()   = quantum defect for n-shells.  non-zero only
C                           for adf04 files with orbital energy data
C                           1st. dim: n-shell (1<=n<=ndqdn)
C  output: (i*4)  iorb    = input data file: number of orbital energies
C
C  output: (i*4)  ia()    = energy level index number
C  output: (c*18) cstrga()= nomenclature/configuration for level 'ia()'
C  output: (i*4)  isa()   = multiplicity for level 'ia()'
C                           note: (isa-1)/2 = quantum number (s)
C  output: (i*4)  ila()   = quantum number (l) for level 'ia()'
C  output: (r*8)  xja()   = quantum number (j-value) for level 'ia()'
C                           note: (2*xja)+1 = statistical weight
C  output: (r*8)  wa()    = energy relative to level 1 (cm-1) for level
C                           'ia()'
C  output: (c*1)  cpla()  = char. specifying 1st parent for level 'ia()'
C                                integer - parent in bwnoa() list
C                                'blank' - parent bwnoa(1)
C                                  'x'   - do not assign a parent
C  output: (i*4)  npla()  = no. of parent/zeta contributions to ionis.
C                           of level
C  output: (i*4)  ipla(,) = parent index for contributions to ionis.
C                           of level
C                           1st dimension: parent index
C                           2nd dimension: level index
C  output: (i*4)  zpla(,) = eff. zeta param. for contributions to ionis.
C                           of level
```

```
C                           1st dimension: parent index
C                           2nd dimension: level index
C
C  output: (i*4)  nv      = input data file: number of gamma/temperature
C                           pairs for a given transition.
C  output: (r*8)  scef()  = input data file: electron temperatures (k)
C                           (initially just the mantissa. see 'itpow()')
C                           (note: te=tp=th is assumed)
C
C  output: (i*4)  itran   = input data file: number of transitions
C  output: (i*4)  maxlev  = highest index level in read transitions
C
C  output: (c*1)  tcode() = transition: data type pointer:
C                           ' ','1','2','3' => elec. impact trans.
C                           'p','P' => proton   impact   transition
C                           'h','H' => charge   exchange recombination
C                           'r','R' => free     electron recombination
C                           'i','I' => coll. ionis. from lower stage ion
C                           's','S' => Ionisation from current ion
C                           'l','L' => L-line for unresolved DR emissivity
C  output: (i*4)  i1a()   = transition:
C                            lower energy level index (case ' ' & 'p')
C                            signed parent index (case 'h','r','s' & 'i')
C  output: (i*4)  i2a()   = transition:
C                            upper energy level index (case ' ' & 'p')
C                            capturing level index (case 'h','r','s' & 'i')
C  output: (r*8)  aval()  = transition:
C                            a-value (sec-1)          (case ' ')
C                            neutral beam energy      (case 'h')
C                            not used             (case 'p','r' & 'i')
C  output: (r*8)  scom(,) = transition:
C                            gamma values             (case ' ' & 'p')
C                            rate coefft.(cm3 sec-1)(case 'h','r' & 'i')
C                            scaled rate coefft.(cm3 sec-1)(case 's')
C                           1st dimension - temperature 'scef()'
C                           2nd dimension - transition number
C  output: (i*4)  beth()  = transition
C                           1st Bethe coefficient     (case ' ','1','2')
C  output: (i*4)  iadftyp = adf04 type: 1=omega, 3=upsilon, 4=non-maxwl.
C  output: (l*4)  lprn    = .true.  => multiple parent data on 1st line
C                         = .false. => multiple parent data not present
C  output: (l*4)  lcpl    = .true.  => parent assignment on level lines
C                         = .false. => parent assignment not present
C  output: (l*4)  lorb    = .true.  => orbital data on level terminator
C                         = .false. => orbital data not present
C  output: (l*4)  lbeth   = .true.  => bethe data on e-transition lines
C                         = .false. => bethe data not present
C  output: (l*4)  letyp   = .true.  => e- excitation transitions present
C  output: (l*4)  lptyp   = .true.  => p- excitation transitions present
C  output: (l*4)  lrtyp   = .true.  => recombination transitions present
C  output: (l*4)  lhtyp   = .true.  => cx transitions present
C  output: (l*4)  lityp   = .true.  => ionis. trans. from z-1 ion present
C  output: (l*4)  lstyp   = .true.  => ionis. trans. from current ion present
C  output: (l*4)  lltyp   = .true.  => 'l'-line for unresolved DR emissivity
C  output: (l*4)  ltied() = .true.  => specified level tied
C                         = .false. => specified level is untied
C                           dimension => level index
C
C          (i*4)  ndmet   = parameter = max. number of metastables allowed
C          (i*4)  ndqdn   = parameter = max. number of n-shells for quantum
C                                       defects
```

```
C          (i*4)  ntdim   = parameter = max. number of internal temperatures
C                                       (must equal nvmax)
C          (r*8)  dzero   = parameter = minimum value for 'aval()' and
C                                       'scom()' arrays = 1.0d-30
C
C          (i*4)  i4unit  = function (see routine selection below)
C          (i*4)  iqs     = x-sect data format selector
C                           note: iqs=3 only allowed in this program
C          (i*4)  ifail   = failure number from xxpars and xxprs1
C          (i*4)  i       = general use.
C          (i*4)  iabt    = return code from 'r(fctn' (0 => no error)
C                           or from interrogation of 'c10'
C          (i*4)  j       = general use.
C          (i*4)  j1      = input data file - selected transition:
C                            lower energy level index (case ' ' & 'p')
C          (i*4)  j2      = input data file - selected transition:
C                            upper energy level index (case ' ' & 'p')
C                            capturing    level index (case 'h' & 'r')
C          (i*4)  lencst  = byte length of string cstrga()
C          (i*4)  iline   = energy level index for current line
C          (i*4)  irecl   = record length of input dataset (<=128)
C          (i*4)  itype   = resolution of parent metastables
C                             1 - ls resolved
C                             2 - lsj resolved
C                             3 - arbitrary resolution
C          (i*4)  iapow   = exponent of 'avalm'
C          (i*4)  igpow() = exponent of 'gamma()'
C          (i*4)  itpow() = temperatures - exponent
C                           note: mantissa initially kept in 'scef()'
C
C          (r*4)  zf      = should be equivalent to 'iz1'
C
C          (r*8)  avalm   = input data file - selected transition:
C                           mantissa of:   ('iapow' => exponent)
C                            a-value (sec-1)          (case ' ')
C                            neutral beam energy      (case 'h')
C                            not used              (case 'p','r','s' & 'i')
C          (r*8)  gamma() = input data file - selected transition:
C                           mantissa of: ('igpow()' => exponent)
C                            gamma values      (case ' ','1','2','3' & 'p')
C                            rate coefft.(cm3 sec-1)(case 'h','r','s' & 'i')
C                           dimension => temperature 'scef()'
C
C          (c*10) c10      = used to parse value for xja()
C          (c*7)  cdelim  = delimiters for input of data from headers
C          (c*25) c25     = used to parse value to cstrga()
C          (c*25) c25t    = copy of c25 : unsatisfactory method of
C                           avoiding compiler reference error :
C                           dhb 07.04.95
C          (c*80) cline   = current energy level index parameter line
C          (c*75) string  = tail string of 1st data line for parsing
C          (c*44) strg1   = tail string of level spec lines for parsing
C          (c*500)buffer  = general string buffer storage
C          (c*3)  citpow()= used to parse values to itpow()
C          (c*5)  cscef() = used to parse values to scef()
C
C          (l*4)  ldata   = identifies  whether  the end of an  input
C                           section in the data set has been located.
C                           (.true. => end of section reached)
C          (l*4)  ltchr   = .true.  => current 'tcode()' = 'h' or 'r'
C                                                          's' or 'i'
```

```
C                        = .false. => current 'tcode()'.ne.'h' or 'r'
C                                                          's' or 'i'
C          (l*4)  ltcpr   = .true.  => current 'tcode()' = 'p' or 'r'
C                                                        's' or 'i'
C                        = .false. => current 'tcode()'.ne. 'p' or 'r'
C                                                          's' or 'i'
C          (l*4) lerror  = .true.  => untied level found
C                        = .false. => all levels tied
C          (l*4)  ltied() = .true.  => specified level tied
C                        = .false. => specified level is untied
C                          dimension => level index
C
C
C note:        ltchr        ltcpr            tcode()
C              ---------------------------------------
C              .true.       .true.      =>   'r','i','s'
C              .true.       .false.     =>   'h'
C              .false.      .true.      =>   'p'
C              .false.      .false.     =>   ' ','1','2','3'
C
C        for a-values & gamma-values entries less than 'dzero' are taken
C        as being equal to dzero. this affects the 'aval()' and 'scom()'
C        arrays.
C
C routines:
C          routine    source    brief description
C          ------------------------------------------------------------
C          xxpars     ADAS      analyses the adf04 1st string for parents
C          xxprs1     ADAS      analyses the adf04 level string for ionis.
C          i4unit     ADAS      fetch unit number for output of messages
C          r8fctn     ADAS      converts from character to real variable
C          i4fctn     ADAS      converts from char. to integer  variable
C          xxslen     ADAS      finds string length excluding leading and
C                               trailing blanks
C          xxword     ADAS      parses a string into separate words
C                               for ' ()<>{}' delimiters
C
C AUTHOR:  Hugh Summers, University of Strathclyde
C          JA7.08
C          tel. 0141-548-4196
C
C DATE:    27/02/03
C
C UPDATE:
C
C VERSION:  1.2
C DATE:     10/09/2004
C
C MODIFIED: Allan Whiteford
C           - Extended code to handle J values greater than 10,000.
C             Actually, to allow greater spacing between the brackets
C             which delimit the J.
C
C VERSION:  1.3
C DATE:     26/11/2004
C
C MODIFIED: Paul Bryans and Allan Whiteford
C           - Fixed some of the comments.
C           - Do not re-order transition indices for type IV file
C           - Upped dimensions to allow 50 energies/temperatures
C
```

```
C VERSION:  1.4
C DATE:     26/11/2004
C
C MODIFIED: Allan Whiteford
C           - Changed dimension checks so that nvmax can be less
C             than ntdim.
C
C VERSION:  1.5
C DATE:     30/11/2004
C
C MODIFIED: Allan Whiteford
C           - Corrected flaw in logic introduced in version 1.4.
C
C VERSION:  1.6
C DATE:     28/07/2008
C
C MODIFIED: Allan Whiteford
C           - Allowed ionisation potential to be zero for bare nuclei
C           - Allowed configurations to be blank (also for bare nuclei)
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      CHARACTER           CPLA(NDLEV)
      CHARACTER*9         CPRTA(NDMET)
      CHARACTER*(*)       CSTRGA(NDLEV)
      CHARACTER           TCODE(NDTRN)
      CHARACTER*3         TITLED
      INTEGER             I1A(NDTRN),  I2A(NDTRN),  IA(NDLEV),   IADFTYP
      INTEGER             IL,          ILA(NDLEV),  IORB
      INTEGER             IPLA(NDMET,NDLEV),        ISA(NDLEV)
      INTEGER             ITIEACTN,    ITRAN,       IUNIT,       IZ
      INTEGER             IZ0,         IZ1,         MAXLEV,      NDLEV
      INTEGER             NDMET,       NDQDN,       NDTRN,       NPL
      INTEGER             NPLA(NDLEV), NV,          NVMAX
      LOGICAL             LBETH,       LBSETA(NDMET),            LCPL
      LOGICAL             LETYP,       LHTYP,       LITYP,       LLTYP
      LOGICAL             LORB,        LPRN,        LPTYP
      LOGICAL             LQDORB((NDQDN*(NDQDN+1))/2),           LRTYP
      LOGICAL             LSTYP,       LTIED(NDLEV)
      REAL*8              AVAL(NDTRN), BETH(NDTRN), BWNO
      REAL*8              BWNOA(NDMET),             PRTWTA(NDMET)
      REAL*8              QDN(NDQDN),  QDORB((NDQDN*(NDQDN+1))/2)
      REAL*8              SCEF(NVMAX), SCOM(NVMAX,NDTRN)
      REAL*8              WA(NDLEV),   XJA(NDLEV),  ZPLA(NDMET,NDLEV)
```

9.96 xxdata_07: Subroutine xxdata_07 from library adaslib

```
       SUBROUTINE XXDATA_07( IUNIT  , DSNAME ,
     &                       NSTORE , NTDIM  ,
     &                       ESYM   , IZ0    ,
     &                       NBSEL  , ISELA  ,
     &                       IZ     , IZ1    ,
     &                       CICODE , CFCODE , CIION , CFION ,
     &                       BWNO   ,
     &                       ITA    ,
     &                       TETA   , SZD
     &                     )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXDATA_07 ******************
C
C  PURPOSE:  TO  FETCH  DATA  FROM  INPUT ELECTRON IMPACT IONIZATION
C            RATE COEFFT FILES (ADF07)
C
C            (MEMBER STORED IN IONELEC.DATA - MEMBER PREFIX 'SZD#').
C
C  CALLING PROGRAM: ADAS502/SSZD
C
C  DATA:
C
C           UP TO 'NSTORE' SETS (DATA-BLOCKS) OF DATA MAY BE  READ FROM
C           THE FILE - EACH BLOCK FORMING A COMPLETE SET OF  IONIZATION
C           RATE COEFFICIENT VALUES FOR GIVEN TEMPERATURES.
C           EACH DATA-BLOCK  IS  ANALYSED INDEPENDENTLY OF ANY  OTHER
C           DATA-BLOCK.
C
C           THE UNITS USED IN THE DATA FILE ARE TAKEN AS FOLLOWS:
C
C           TEMPERATURES        : EV
C           RATE COEFFT         : CM**3 SEC-1
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  IUNIT    = UNIT TO WHICH INPUT FILE IS ALLOCATED.
C
C  INPUT : (I*4)  NSTORE   = MAXIMUM NUMBER  OF  INPUT DATA-BLOCKS  THAT
C                            CAN BE STORED.
C  INPUT : (I*4)  NTDIM    = MAX NUMBER OF ELECTRON TEMPERATURES ALLOWED
C
C  OUTPUT: (C*2)  ESYM     = READ - IONISING ION - ELEMENT SYMBOL
C  OUTPUT: (I*4)  IZ0      = READ - IONISING ION - NUCLEAR CHARGE
C
C  OUTPUT: (I*4)  NBSEL    = NUMBER OF DATA-BLOCKS ACCEPTED & READ IN.
C  OUTPUT: (I*4)  ISELA()  = READ - DATA-SET DATA-BLOCK ENTRY INDICES
C                            DIMENSION: DATA-BLOCK INDEX
C
C  OUTPUT: (I*4)  IZ()     = READ - IONISING ION - INITIAL CHARGE
C                            DIMENSION: DATA-BLOCK INDEX
C  OUTPUT: (I*4)  IZ1()    = READ - IONISING ION - FINAL   CHARGE
C                            DIMENSION: DATA-BLOCK INDEX
C
C  OUTPUT: (C*2)  CICODE() = READ - INITIAL STATE METASTABLE INDEX
C                            DIMENSION: DATA-BLOCK INDEX
C  OUTPUT: (C*2)  CFCODE() = READ - FINAL   STATE METASTABLE INDEX
C                            DIMENSION: DATA-BLOCK INDEX
C  OUTPUT: (C*5)  CIION()  = READ - INITIAL ION (as <ESYM>+(IZ()> )
```

```
C                               DIMENSION: DATA-BLOCK INDEX
C  OUTPUT: (C*5)  CFION()   = READ - FINAL   ION (as <ESYM>+<IZ1()>)
C                               DIMENSION: DATA-BLOCK INDEX
C
C  OUTPUT: (R*8)  BWNO()    = READ- EFFECTIVE IONIZATION POTENTIAL (CM-1)
C                               DIMENSION: DATA-BLOCK INDEX
C
C  OUTPUT: (I*4)  ITA()     = READ - NUMBER OF ELECTRON TEMPERATURES
C                               DIMENSION: DATA-BLOCK INDEX
C
C  OUTPUT: (R*8)  TETA(,)   = READ - ELECTRON TEMPERATURES (UNITS: eV)
C                               1st DIMENSION: ELECTRON TEMPERATURE INDEX
C                               2nd DIMENSION: DATA-BLOCK INDEX
C
C  OUTPUT: (R*8)  SZD(,)     =READ - FULL SET OF ZERO DENSITY IONIZATION
C                                    RATE COEFFICIENT VALUES (cm**3/sec)
C                               1st DIMENSION: ELECTRON TEMPERATURE INDEX
C                               2nd DIMENSION: DATA-BLOCK INDEX
C
C          (I*4)  I4EIZ0    = FUNCTION - (SEE ROUTINES SECTION BELOW)
C          (I*4)  I4FCTN    = FUNCTION - (SEE ROUTINES SECTION BELOW)
C          (I*4)  I4UNIT    = FUNCTION - (SEE ROUTINE SECTION BELOW)
C          (I*4)  IBLK      = ARRAY INDEX: DATA-BLOCK INDEX
C          (I*4)  ITT       = ARRAY INDEX: ELECTRON TEMPERATURE INDEX
C          (I*4)  NTNUM     = NUMBER OF ELECTRON TEMPERATURES FOR CURRENT
C                               DATA-BLOCK
C          (I*4)  IABT      = RETURN CODE FROM 'I4FCTN'
C          (I*4)  IPOS1     = GENERAL USE STRING INDEX VARIABLE
C          (I*4)  IPOS2     = GENERAL USE STRING INDEX VARIABLE
C
C          (R*8)  R8FCTN    = FUNCTION - (SEE ROUTINES SECTION BELOW)
C
C          (L*4)  LBEND     = IDENTIFIES WHETHER THE LAST OF THE  INPUT
C                               DATA SUB-BLOCKS HAS BEEN LOCATED.
C                               (.TRUE. => END OF SUB-BLOCKS REACHED)
C
C          (C*1)  CSLASH    = '/' - DELIMITER FOR 'XXHKEY'
C          (C*2)  C2        = GENERAL USE TWO BYTE CHARACTER STRING
C          (C*4)  CKEY1     = 'I.P.'  - INPUT BLOCK HEADER KEY
C          (C*5)  CKEY2     = 'ICODE' - INPUT BLOCK HEADER KEY
C          (C*5)  CKEY3     = 'FCODE' - INPUT BLOCK HEADER KEY
C          (C*4)  CKEY4     = 'ISEL'  - INPUT BLOCK HEADER KEY
C          (C*10) C10       = GENERAL USE TEN BYTE CHARACTER STRING
C          (C*80) C80       = GENERAL USE 80 BYTE  CHARACTER  STRING  FOR
C                               THE INPUT OF DATA-SET RECORDS.
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          XXHKEY     ADAS      OBTAIN KEY/RESPONSE STRINGS FROM TEXT
C          I4EIZ0     ADAS      INTEGER*4 FUNCTION    -
C                               RETURNS Z0 FOR GIVEN ELEMENT SYMBOL
C          I4FCTN     ADAS      INTEGER*4 FUNCTION    -
C                               CONVERT CHARACTER STRING TO INTEGER
C          I4UNIT     ADAS      INTEGER*4 FUNCTION    -
C                               FETCH UNIT NUMBER FOR OUTPUT OF MESSAGES
C          R8FCTN     ADAS      REAL*8 FUNCTION       -
C                               CONVERT CHARACTER STRING TO REAL*8
C          XXCASE     ADAS      SWITCHES CASE
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
```

```
C          K1/0/37
C          JET EXT. 2520
C
C DATE:    07/06/91
C
C UPDATE:  23/04/93 - PE BRIDEN - ADAS91: ADDED I4UNIT FUNCTION TO WRITE
C                                         STATEMENTS FOR SCREEN MESSAGES
C
C UPDATE:  24/05/93 - PE BRIDEN - ADAS91: CHANGED I4UNIT(0)-> I4UNIT(-1)
C
C UPDATE:  10/11/94 - L. JALOTA - MODIFIED TO RUN UNDER UNIX.
C
C-----------------------------------------------------------------------
C
C NOTES: Copied from e2data.for. This is v1.1 of xxdata_07.
C
C VERSION  : 1.1
C DATE     : 26-03-2008
C MODIFIED : Allan Whiteford
C              - First version
C
C VERSION  : 1.2
C DATE     : 21-05-2008
C MODIFIED : Martin O'Mullane
C              - Permit lower case datasets for 2008 onwards data.
C              - Extract iz0 from element symbol in first block.
C
C-----------------------------------------------------------------------
      CHARACTER*2         CFCODE(NSTORE)
      CHARACTER*5         CFION(NSTORE)
      CHARACTER*2         CICODE(NSTORE)
      CHARACTER*5         CIION(NSTORE)
      CHARACTER*80        DSNAME
      CHARACTER*2         ESYM
      INTEGER             ISELA(NSTORE),            ITA(NSTORE), IUNIT
      INTEGER             IZ(NSTORE),  IZ0,         IZ1(NSTORE), NBSEL
      INTEGER             NSTORE,      NTDIM
      REAL*8              BWNO(NSTORE),             SZD(NTDIM,NSTORE)
      REAL*8              TETA(NTDIM,NSTORE)
```

9.97 xxdata_08: Subroutine xxdata_08 from library adaslib

```
      SUBROUTINE XXDATA_08( IUNIT  , NDPRT  , NDLEV , NDT   ,
     &                      SEQSYM , IZ     , IZ0   , IZ1   ,
     &                      NPRNT  , NPRNTI , BWNP  ,
     &                      IPA    , CSTRPA , ISPA  , ILPA  , XJPA  ,
     &                      WPA    ,
     &                      IL     , BWNR   ,
     &                      IA     , CSTRGA , ISA   , ILA   , XJA   ,
     &                      WA     ,
     &                      IPRTI  , TPRTI  , ISPRTI, RADR  , LRADR ,
     &                      NTE    , TEA
     &                    )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXDATA_08 ******************
C
C  PURPOSE:  TO FETCH DATA FROM INPUT ADF08 DATA SET.
C
C  CALLING PROGRAM: ADAS411
C
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  IUNIT   = UNIT TO WHICH INPUT FILE IS ALLOCATED
C
C  INPUT : (I*4)  NDPRT   = MAXIMUM NUMBER OF PARENT STATES
C  INPUT : (I*4)  NDLEV   = MAXIMUM NUMBER OF RESOLVED LEVELS
C  INPUT : (I*4)  NDT     = MAX. NUMBER OF ELECTRON TEMPERATURES
C
C  OUTPUT: (C*2)  SEQSYM  = RECOMBINED ION SEQ
C  OUTPUT: (I*4)  IZ      = RECOMBINED ION CHARGE
C  OUTPUT: (I*4)  IZ0     = NUCLEAR CHARGE
C  OUTPUT: (I*4)  IZ1     = RECOMBINING ION CHARGE
C  OUTPUT: (I*4)  NPRNT   = TOTAL NUMBER OF PARENTS
C  OUTPUT: (I*4)  NPRNTI  = NUMBER OF PARENTS WHICH ARE INITIAL PARENTS
C  OUTPUT: (R*8)  BWNP    = BINDING WAVE NO. OF GROUND PARENT (CM-1)
C  OUTPUT: (I*4)  IPA()   = NUMBER OF PARENT ENERGY LEVELS
C  OUTPUT: (C*18) CSTRPA()= NOMENCL./CONFIG. FOR PARENT LEVEL 'IPA()'
C  OUTPUT: (I*4)  ISPA()  = MULTIPLICITY FOR PARENT LEVEL 'IPA()'
C                           NOTE: (ISPA-1)/2 = QUANTUM NUMBER (SP)
C  OUTPUT: (I*4)  ILPA()  = QUANTUM NUMBER (LP) FOR PARENT LEVEL 'IPA()'
C  OUTPUT: (R*8)  XJPA()  = QUANTUM NUMBER (JP) FOR PARENT LEVEL 'IPA()'
C                           NOTE: (2*XJPA)+1 = STATISTICAL WEIGHT
C  OUTPUT: (R*8)  WPA()   = ENERGY RELATIVE TO PARENT LEVEL 1 (CM-1)
C                           FOR PARENT LEVEL 'IPA()'
C
C  OUTPUT: (I*4)  IL      = NUMBER OF ENERGY LEVELS (TERMS) OF
C                           RECOMBINED ION
C  OUTPUT: (R*8)  BWNR    = IONISATION POTENTIAL (CM-1) OF LOWEST LEVEL
C                           OF RECOMBINED ION
C  OUTPUT: (I*4)  IA()    = RECOMBINED ION ENERGY LEVEL INDEX NUMBER
C  OUTPUT: (C*18) CSTRGA()= NOMENCL./CONFIG. FOR RECOMBINED ION LEVEL
C                           'IA()'
C  OUTPUT: (I*4)  ISA()   = MULTIPLICITY FOR RECOMBINED LEVEL 'IA()'
C                           NOTE: (ISA-1)/2 = QUANTUM NUMBER (S)
C  OUTPUT: (I*4)  ILA()   = QUANTUM NUMBER (L) FOR RECOMBINED LEVEL
C                           'IA()'
C  OUTPUT: (R*8)  XJA()   = QUANTUM NUMBER (J) FOR RECOMBINED LEVEL
C                           'IA()'
C                           NOTE: (2*XJA)+1 = STATISTICAL WEIGHT
```

```
C  OUTPUT: (R*8)  WA()    = ENERGY RELATIVE TO RECOMBINED LEVEL 1 (CM-1)
C                          FOR RECOMBINED LEVEL 'IA()'
C  OUTPUT: (I*4)  IPRTI() = INITIAL PARENT BLOCK INDEX
C  OUTPUT: (C*5)  TPRTI() = INITIAL PARENT BLOCK TERM
C  OUTPUT: (I*4)  ISPRTI()= INITIAL PARENT BLOCK SPIN MULTIPLICITY
C  OUTPUT: (R*8)  TEA()   = ELECTRON TEMPERATURES (K)
C  OUTPUT: (R*8)  RADR(,,)= TERM SELECTIVE DIELEC. COEFFTS.(CM3 S-1)
C                          1ST.DIM: LEVEL INDEX
C                          2ND.DIM: INITIAL PARENT INDEX
C                          3RD.DIM: TEMPERATURE INDEX
C  OUTPUT: (L*4)  LRADR(,)= .TRUE. => DIEL. PRESENT FOR LEVEL INDEX
C                          .FALSE.=> DIEL. NOT PRESENT FOR LEVEL INDEX
C                          1ST.DIM: LEVEL INDEX
C                          2ND.DIM: INITIAL PARENT INDEX
C
C          (I*4)  INDX    = GENERAL INDEX
C          (I*4)  II      = GENERAL INDEX
C          (I*4)  I       = GENERAL INDEX
C          (I*4)  IPI     = GENERAL INDEX
C          (I*4)  IPF     = GENERAL INDEX
C          (I*4)  IPFS    = GENERAL INDEX
C          (I*4)  J       = GENERAL INDEX
C          (I*4)  K       = GENERAL INDEX
C
C          (L)    LDATA   = GENERAL READ/DO NOT READ FLAG
C          (L)    LNOPI   = FLAG TO DETERMINE WHETHER HAVE PASSED
C     INTO A NEW INITIAL PARENT BLOCK
C
C          (C*20) C20     = GENERAL CHARACTER STRING
C
C  ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4UNIT     ADAS      FETCH UNIT NUMBER FOR OUTPUT OF MESSAGES
C          I4EIZ0     ADAS      RETURNS NUCL. CHARGE FROM ELEMENT SYMBOL
C          R8FCTN     ADAS      CONVERTS FROM CHARACTER TO REAL VARIABLE
C          XXWORD     ADAS      EXTRACT POSITION OF NUMBER IN BUFFER
C
C  AUTHOR: H. P. SUMMERS, UNIVERSITY OF STRATHCLYDE
C          JA8.08
C          TEL. 0141-553-4196
C
C  DATE:    10/11/97
C
C  UPDATE:
C
C VERSION: 1.1 DATE: 10-03-98
C MODIFIED: RICHARD MARTIN
C - PUT UNDER SCCS CONTROL
C
C-----------------------------------------------------------------------
C
C NOTES: Copied from dbdata.for. This is v1.1 of xxdata_08.
C
C VERSION  : 1.1
C DATE     : 27-03-2008
C MODIFIED : Allan Whiteford
C               - First version
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
```

```
CHARACTER*(*)       CSTRGA(NDLEV),            CSTRPA(NDPRT)
CHARACTER*2         SEQSYM
CHARACTER*5         TPRTI(NDPRT)
INTEGER             IA(NDLEV),   IL,          ILA(NDLEV)
INTEGER             ILPA(NDPRT), IPA(NDPRT),  IPRTI(NDPRT)
INTEGER             ISA(NDLEV),  ISPA(NDPRT), ISPRTI(NDPRT)
INTEGER             IUNIT,       IZ,          IZ0,         IZ1
INTEGER             NDLEV,       NDPRT,       NDT,         NPRNT
INTEGER             NPRNTI,      NTE
LOGICAL             LRADR(NDLEV,NDPRT)
REAL*8              BWNP,        BWNR
REAL*8              RADR(NDLEV,NDPRT,NDT),    TEA(NDT)
REAL*8              WA(NDLEV),   WPA(NDPRT),  XJA(NDLEV)
REAL*8              XJPA(NDPRT)
```

9.98 xxdata_09: Subroutine xxdata_09 from library adaslib

```
      SUBROUTINE xxdata_09( IUNIT  , NDPRT  , NDREP , NDLEV ,
     &                      NDAUG  , NDT   ,
     &                      SEQSYM , IZ     , IZ0   , IZ1   ,
     &                      NPRNT  , NPRNTI , NPRNTF, BWNP  ,
     &                      IPA    , CSTRPA , ISPA  , ILPA  , XJPA  ,
     &                      WPA    ,
     &                      IL     , BWNR   ,
     &                      IA     , CSTRGA , ISA   , ILA   , XJA   ,
     &                      WA     ,
     &                      NREP   , IAPRS  , CAPRS , IPAUG ,
     &                      IREPA  , NREPA  , AUGA  , LAUGA ,
     &                      IPRTI  , TPRTI  , ISPRTI, DIELR , LDIELR,
     &                      IPRTF  , TPRTF  , ISPRTF,
     &                      NSYSF  , ISYS   , ISPSYS, DIELN , LDIELN,
     &                      DIELT  ,
     &                      NTE    , TEA    , AATP ,  DTOT
     &                    )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: DADATA *********************
C
C  PURPOSE:  TO FETCH DATA FROM INPUT ADF09 DATA SET.
C
C  CALLING PROGRAM:
C
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  IUNIT   = UNIT TO WHICH INPUT FILE IS ALLOCATED
C
C  INPUT : (I*4)  NDPRT   = MAXIMUM NUMBER OF PARENT STATES
C  INPUT : (I*4)  NDREP   = MAX. NUMBER OF REPRESENTATIVE N-SHELLS
C  INPUT : (I*4)  NDLEV   = MAXIMUM NUMBER OF RESOLVED LEVELS
C  INPUT : (I*4)  NDAUG   = MAXIMUM NUMBER OF AUGER RATE INITIAL AND
C                           FINAL PARENT PAIRS
C  INPUT : (I*4)  NDT     = MAX. NUMBER OF ELECTRON TEMPERATURES
C
C  OUTPUT: (C*2)  SEQSYM  = RECOMBINED ION SEQ
C  OUTPUT: (I*4)  IZ      = RECOMBINED ION CHARGE
C  OUTPUT: (I*4)  IZ0     = NUCLEAR CHARGE
C  OUTPUT: (I*4)  IZ1     = RECOMBINING ION CHARGE
C  OUTPUT: (I*4)  NPRNT   = TOTAL NUMBER OF PARENTS
C  OUTPUT: (I*4)  NPRNTI  = NUMBER OF PARENTS WHICH ARE INITIAL PARENTS
C  OUTPUT: (I*4)  NPRNTF  = NUMBER OF PARENTS WHICH ARE FINAL PARENTS
C  OUTPUT: (R*8)  BWNP    = BINDING WAVE NO. OF GROUND PARENT (CM-1)
C  OUTPUT: (I*4)  IPA()   = NUMBER OF PARENT ENERGY LEVELS
C  OUTPUT: (C*18) CSTRPA()= NOMENCL./CONFIG. FOR PARENT LEVEL 'IPA()'
C  OUTPUT: (I*4)  ISPA()  = MULTIPLICITY FOR PARENT LEVEL 'IPA()'
C                           NOTE: (ISPA-1)/2 = QUANTUM NUMBER (SP)
C  OUTPUT: (I*4)  ILPA()  = QUANTUM NUMBER (LP) FOR PARENT LEVEL 'IPA()'
C  OUTPUT: (R*8)  XJPA()  = QUANTUM NUMBER (JP) FOR PARENT LEVEL 'IPA()'
C                           NOTE: (2*XJPA)+1 = STATISTICAL WEIGHT
C  OUTPUT: (R*8)  WPA()   = ENERGY RELATIVE TO PARENT LEVEL 1 (CM-1)
C                           FOR PARENT LEVEL 'IPA()'
C
C  OUTPUT: (I*4)  IL      = NUMBER OF ENERGY LEVELS (TERMS) OF
C                           RECOMBINED ION
C  OUTPUT: (R*8)  BWNR    = IONISATION POTENTIAL (CM-1) OF LOWEST LEVEL
C                           OF RECOMBINED ION
```

```
C  OUTPUT: (I*4)  IA()    = RECOMBINED ION ENERGY LEVEL INDEX NUMBER
C  OUTPUT: (I*4)  IP()    = ???
C  OUTPUT: (C*18) CSTRGA()= NOMENCL./CONFIG. FOR RECOMBINED ION LEVEL
C                           'IA()'
C  OUTPUT: (I*4)  ISA()   = MULTIPLICITY FOR RECOMBINED LEVEL 'IA()'
C                           NOTE: (ISA-1)/2 = QUANTUM NUMBER (S)
C  OUTPUT: (I*4)  ILA()   = QUANTUM NUMBER (L) FOR RECOMBINED LEVEL
C                           'IA()'
C  OUTPUT: (R*8)  XJA()   = QUANTUM NUMBER (J) FOR RECOMBINED LEVEL
C                           'IA()'
C                           NOTE: (2*XJA)+1 = STATISTICAL WEIGHT
C  OUTPUT: (R*8)  WA()    = ENERGY RELATIVE TO RECOMBINED LEVEL 1 (CM-1)
C                           FOR RECOMBINED LEVEL 'IA()'
C  OUTPUT: (I*4)  NREP    = NUMBER OF REPRESENTATIVE N-SHELLS
C  OUTPUT: (I*4)  IREPA() = REPRESENTATIVE N-SHELL INDEX NUMBER
C  OUTPUT: (I*4)  NREPA() = REPRESENTATIVE N-SHELLS
C  OUTPUT: (I*4)  IAPRS   = NUMBER OF AUGER RATE INITIAL AND FINAL
C                           PARENT PAIRS
C  OUTPUT: (C*10) CAPRS() = AUGER RATE PARENT PAIR STRING
C                           1ST.DIM: PARENT PAIR INDEX
C  OUTPUT: (I*40) IPAUG(,)= INITIAL AND FINAL PARENTS FOR AUGER BREAKUPS
C                           1ST.DIM: PARENT PAIR INDEX
C                           2ND.DIM: INITIAL AND FINAL PARENT INDICES
C  OUTPUT: (R*8)  AUGA(,) = AUGER RATES (SEC-1)
C                           1ST.DIM: REPRESENTATIVE N-SHELL INDEX
C                           2ND.DIM: PARENT PAIR INDEX
C  OUTPUT: (L*4)  LAUGA(,) = .TRUE. => AUGER RATE PRESENT FOR N-SHELL
C                            .FALSE.=> AUGER RATE NOT PRESENT
C                           1ST.DIM: REPRESENTATIVE N-SHELL INDEX
C                           2ND.DIM: PARENT PAIR INDEX
C  OUTPUT: (I*4)  IPRTI() = INITIAL PARENT BLOCK INDEX
C  OUTPUT: (C*5)  TPRTI() = INITIAL PARENT BLOCK TERM
C  OUTPUT: (I*4)  ISPRTI()= INITIAL PARENT BLOCK SPIN MULTIPLICITY
C  OUTPUT: (R*8)  TEA()   = ELECTRON TEMPERATURES (K)
C  OUTPUT: (R*8)  DIELR(,,)= TERM SELECTIVE DIELEC. COEFFTS.(CM3 S-1)
C                           1ST.DIM: LEVEL INDEX
C                           2ND.DIM: INITIAL PARENT INDEX
C                           3RD.DIM: TEMPERATURE INDEX
C  OUTPUT: (L*4)  LDIELR(,)= .TRUE. => DIEL. PRESENT FOR LEVEL INDEX
C                            .FALSE.=> DIEL. NOT PRESENT FOR LEVEL INDEX
C                           1ST.DIM: LEVEL INDEX
C                           2ND.DIM: INITIAL PARENT INDEX
C  OUTPUT: (I*4)  IPRTF(,) = FINAL PARENT BLOCK INDEX
C  OUTPUT: (C*5)  TPRTF(,) = FINAL PARENT BLOCK TERM
C  OUTPUT: (I*4)  ISPRTF(,)= FINAL PARENT BLOCK SPIN MULTIPLICITY
C  OUTPUT: (I*4)  NSYSF(,) = NO,. OF SPIN SYSTEMS BUILT ON FINAL PARENT
C  OUTPUT: (I*4)  ISYS(,,) = N-SHELL SPIN SYSTEM INDEX FOR FINAL PARENT
C  OUTPUT: (I*4)  ISPSYS(,,)=N-SHELL SPIN SYSTEM FOR FINAL PARENT
C  OUTPUT: (R*8)  DIELN(,,,,) =N-SHELL DIELEC. COEFFTS.(CM3 S-1)
C                           1ST.DIM: REPR. N-SHELL INDEX
C                           2ND.DIM: INITIAL PARENT INDEX
C                           3RD.DIM: FINAL PARENT INDEX
C                           4TH.DIM: SPIN SYSTEM INDEX
C                           5TH.DIM: TEMPERATURE INDEX
C  OUTPUT: (R*8)  LDIELN(,)= .TRUE. => DIEL. PRESENT FOR REPR. N-SHELL
C                            .FALSE.=> DIEL. NOT PRESENT FOR  N-SHELL
C                           1ST.DIM: REPR. N-SHELL INDEX
C                           2ND.DIM: INITIAL PARENT INDEX
C                           3RD.DIM: FINAL PARENT INDEX
C                           4TH.DIM: SPIN SYSTEM INDEX
C  OUTPUT: (R*8)  DIELT(,,,) =N-SHELL DIELEC. COEFFTS.(CM3 S-1)
```

```
C                           1ST.DIM: INITIAL PARENT INDEX
C                           2ND.DIM: FINAL PARENT INDEX
C                           3RD.DIM: SPIN SYSTEM INDEX
C                           4TH.DIM: TEMPERATURE INDEX
C  OUTPUT: (R*8)  AATP()   =SPECIFIC METASTABLE TO METASTABLE SEC.
C                            AUGER PATHS FOR TERMS
C  OUTPUT: (R*8)  DTOT(,,,) =TOTAL DR COEFFICINETS (CM3 S-1) TABULATED
C                           AT BOTTOM OF FILE
C                           1ST.DIM: INITIAL PARENT INDEX
C                           2ND.DIM: ELECTRON TEMPERATURE INDEX
C
C          (I*4)  INDX    = GENERAL INDEX
C          (I*4)  INDX1   = GENERAL INDEX
C          (I*4)  II      = GENERAL INDEX
C          (I*4)  I       = GENERAL INDEX
C          (I*4)  IPI     = GENERAL INDEX
C          (I*4)  IPF     = GENERAL INDEX
C          (I*4)  IPFS    = GENERAL INDEX
C          (I*4)  J       = GENERAL INDEX
C          (I*4)  K       = GENERAL INDEX
C          (I*4)  IFORM   = FORMAT OF FILE:
C                           1: LS 93 DATA
C                           2: LS 00 DATA
C                           3: IC 00 DATA
C
C          (L)    LDATA   = GENERAL READ/DO NOT READ FLAG
C          (L)    LNOPI   = FLAG TO DETERMINE WHETHER HAVE PASSED
C     INTO A NEW INITIAL PARENT BLOCK
C
C          (C*20) C20     = GENERAL CHARACTER STRING
C
C  ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4UNIT     ADAS      FETCH UNIT NUMBER FOR OUTPUT OF MESSAGES
C          I4EIZ0     ADAS      RETURNS NUCL. CHARGE FROM ELEMENT SYMBOL
C          R8FCTN     ADAS      CONVERTS FROM CHARACTER TO REAL VARIABLE
C          XXWORD     ADAS      EXTRACT POSITION OF NUMBER IN BUFFER
C          DASUMD     ADAS      SUM DR OVER REPRESANTITIVE N-SHELLS
C
C
C  BASED ON DADATA V1.2
C
C  AUTHOR:  A D WHITEFORD, UNIVERSITY OF STRATHCLYDE
C
C  DATE:    02/05/05
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      CHARACTER*10        CAPRS(NDAUG)
      CHARACTER*(*)       CSTRGA(NDLEV),            CSTRPA(NDPRT)
      CHARACTER*2         SEQSYM
      CHARACTER*5         TPRTF(NDPRT,NDPRT),       TPRTI(NDPRT)
      INTEGER             IA(NDLEV),   IAPRS,       IL
      INTEGER             ILA(NDLEV),  ILPA(NDPRT), IPA(NDPRT)
      INTEGER             IPAUG(NDAUG,2),           IPRTF(NDPRT,NDPRT)
      INTEGER             IPRTI(NDPRT),             IREPA(NDREP)
      INTEGER             ISA(NDLEV),  ISPA(NDPRT), ISPRTF(NDPRT,NDPRT)
      INTEGER             ISPRTI(NDPRT)
      INTEGER             ISPSYS(NDPRT,NDPRT,2),    ISYS(NDPRT,NDPRT,2)
      INTEGER             IUNIT,       IZ,          IZ0,         IZ1
```

```
INTEGER             NDAUG,       NDLEV,       NDPRT,       NDREP
INTEGER             NDT,         NPRNT,       NPRNTF,      NPRNTI
INTEGER             NREP,        NREPA(NDREP)
INTEGER             NSYSF(NDPRT,NDPRT),       NTE
LOGICAL             LAUGA(NDREP,NDAUG)
LOGICAL             LDIELN(NDREP,NDPRT,NDPRT,2)
LOGICAL             LDIELR(NDLEV,NDPRT)
REAL*8              AATP(NDLEV,NDPRT),        AUGA(NDREP,NDAUG)
REAL*8              BWNP,        BWNR
REAL*8              DIELN(NDREP,NDPRT,NDPRT,2,NDT)
REAL*8              DIELR(NDLEV,NDPRT,NDT)
REAL*8              DIELT(NDPRT,NDPRT,2,NDT), DTOT(NDPRT,NDT)
REAL*8              TEA(NDT),    WA(NDLEV),   WPA(NDPRT)
REAL*8              XJA(NDLEV),  XJPA(NDPRT)
```

9.99 xxdata_11: Subroutine xxdata_11 from library adaslib

```
      subroutine  xxdata_11( iunit  , iclass ,
     &                       isdimd , iddimd , itdimd ,
     &                       ndptnl , ndptn  , ndptnc , ndcnct ,
     &                       iz0    , is1min , is1max ,
     &                       nptnl  , nptn   , nptnc  ,
     &                       iptnla , iptna  , iptnca ,
     &                       ncnct  , icnctv ,
     &                       iblmx  , ismax  , dnr_ele, dnr_ams,
     &                       isppr  , ispbr  , isstgr ,
     &                       idmax  , itmax  ,
     &                       ddens  , dtev   , drcof  ,
     &                       lres   , lstan  , lptn
     &                     )
c-----------------------------------------------------------------------
c
c  ****************** fortran77 subroutine: xxdata_11 ******************
c
c  purpose: to read a complete  adf11 file, check its class and
c           determine its standard, resolved and partition organisation.
c
c  calling program: various
c
c  notes:    (1) A 'standard' adf11 file contains gcr data between one
c                whole ionisation stage and another whole ionisation
c                stage.
c                A 'resolved' (or partial) adf11 file contains gcr data
c                between a set of metastables of one ionisation stage
c                and a set of metastables of another ionisation stage.
c                A resolved file is distinguished from a standard file
c                by the presence of a 'connection vector' in the adf11
c                data file header lines.
c                The connection vector specifies the number of meta-
c                stables in each ionisation stage which are coupled
c                together by gcr data.
c            (2) A 'partitioned' adf11 file contains gcr data between
c                clumps of ionisation stages or metastables or comb-
c                inations of the two called 'partitions'.
c                A 'partition level' is a specification of the
c                partitions which span all the ionisation stages (and
c                metastables) of an element.  Successive partition
c                levels give a heirarchy corresponding to larger
c                partitions and greater clumping.
c                A 'superstage' is a set of partitions which are close-
c                coupled.
c                There are thus equivalences :
c                        ionisation stage   -   superstage
c                        metastable         -   partition
c                        ion charge         -   superstage index
c                A partitioned adf11 file may be standard (with each
c                superstage comprising only one partition) or resolved.
c                A partitioned file is distinguished by the presence of
c                'partition specification block' in the adf11 data
c                file header lines.
c            (3) When  a partition specification block is present, it
c                should be ordered from the highest partition level
c                index to lowest partition level index.  Thus the first
c                partition in the partition block has the least number
c                of partitions and the last has the greatest number.
c            (4) Twelev classes of adf11 data file may be read by the
```

```
c                subroutine as follow:
c
c              class index type   GCR data content
c              ---------------  ----------------
c                  1   acd   recombination coeffts
c                  2   scd   ionisation coeffts
c                  3   ccd   CX recombination coeffts
c                  4   prb   recomb/brems power coeffts
c                  5   prc   CX power coeffts
c                  6   qcd   base meta. coupl. coeffts
c                  7   xcd   parent meta. coupl. coeffts
c                  8   plt   low level line power coeffts
c                  9   pls   represent. line power coefft
c                 10   zcd   effective charge
c                 11   ycd   effective squared charge
c                 12   ecd   effective ionisation potential
c
c             (5) A resolved adf11 file, with a connection vector, has a set
c                 of names and pointers at precise positions in the data file
c                 which are recognised.
c                 The names are different for partitioned and unpartitioned
c                 data files as follow:
c
c                       file      unpartitioned         partitioned
c                       class        names                 names
c
c                       (all)         z1                   s1
c
c                         (indices 1 and 2)     (indices 1 and 2)
c                    ----   ----     ----         ----  ----
c                    acd        iprt     igrd         ispp   ispb
c                    scd        iprt     igrd         ispp   ispb
c                    ccd        iprt     igrd         ispp   ispb
c                    prb        iprt                  ispp
c                    prc        iprt                  ispp
c                    qcd        igrd     jgrd         ispb   jspb
c                    xcd        iprt     jprt         ispp   jspp
c                    plt        igrd           ispb
c                    pls        igrd           ispb
c                    zcd        igrd           ispb
c                    ycd        igrd           ispb
c                    ecd        igrd           ispb
c
c              (6) In partitioned nomenclature: s=superstage; p=partition;
c                  b=base (current superstage), p=parent (next up super-
c                  stage), c=child (next down superstage). Thus arrays
c                  'iprtr' and 'igrd' in old notation are now substituted
c                  by 'isppr' and 'ispbr' respectively internally and in
c                  external naming.
c
c
c  subroutine:
c
c  input : (i*4)  iunit     = unit to which input file is allocated
c  input : (i*4)  iclass    = class of data (1 - 12 ):
c                               1-acd, 2-scd, 3-ccd, 4-prb, 5-prc
c                               6-qcd, 7-xcd, 8-plt, 9-pls,10-zcd
c                              11-ycd,12-ecd
c
c  input : (i*4)  isdimd    = maximum number of (sstage, parent, base)
c                             blocks in isonuclear master files
```

```
c  input : (i*4)  iddimd    = maximum number of dens values in
c                             isonuclear master files
c  input : (i*4)  itdimd    = maximum number of temp values in
c                             isonuclear master files
c  input : (i*4)  ndptnl    = maximum level of partitions
c  input : (i*4)  ndptn     = maximum no. of partitions in one level
c  input : (i*4)  ndptnc    = maximum no. of components in a partition
c  input : (i*4)  ndcnct    = maximum number of elements in connection
c                             vector
c
c  output: (i*4)  iz0       = nuclear charge
c  output: (i*4)  is1min    = minimum ion charge + 1
c                             (generalised to connection vector index)
c  output: (i*4)  is1max    = maximum ion charge + 1
c                             (note excludes the bare nucleus)
c                             (generalised to connection vector index
c                              and excludes last one which always remains
c                              the bare nucleus)
c  output: (i*4)  nptnl     = number of partition levels in block
c  output: (i*4)  nptn()    = number of partitions in partition level
c                             1st dim: partition level
c  output: (i*4)  nptnc(,)  = number of components in partition
c                             1st dim: partition level
c                             2nd dim: member partition in partition level
c  output: (i*4)  iptnla()  = partition level label (0=resolved root,1=
c                                                    unresolved root)
c                             1st dim: partition level index
c  output: (i*4)  iptna(,)  = partition member label (labelling starts at 0)
c                             1st dim: partition level index
c                             2nd dim: member partition index in partition
c                             level
c  output: (i*4)  iptnca(,,)= component label (labelling starts at 0)
c                             1st dim: partition level index
c                             2nd dim: member partition index in partition
c                             level
c                             3rd dim: component index of member partition
c  output: (i*4)  ncnct     = number of elements in connection vector
c  output: (i*4)  icnctv()  = connection vector of number of partitions
c                             of each superstage in resolved case
c                             including the bare nucleus
c                             1st dim: connection vector index
c
c  output: (i*4)  iblmx     = number of (sstage, parent, base)
c                             blocks in isonuclear master file
c  output: (i*4)  ismax     = number of charge states
c                             in isonuclear master file
c                             (generalises to number of elements in
c                              connection vector)
c  output: (c*12) dnr_ele   = CX donor element name for iclass = 3 or 5
c                             (blank if unset)
c  output: (r*8)  dnr_ams   = CX donor element mass for iclass = 3 or 5
c                             (0.0d0 if unset)
c  output: (i*4)  isppr()   = 1st (parent) index for each partition block
c                             1st dim: index of (sstage, parent, base)
c                                      block in isonuclear master file
c  output: (i*4)  ispbr()   = 2nd (base) index for each partition block
c                             1st dim: index of (sstage, parent, base)
c                                      block in isonuclear master file
c  output: (i*4)  isstgr()  = s1 for each resolved data block
c                             (generalises to connection vector index)
c                             1st dim: index of (sstage, parent, base)
```

```
c                                       block in isonuclear master file
c
c  output: (i*4)  idmax     = number of dens values in
c                             isonuclear master files
c  output: (i*4)  itmax     = number of temp values in
c                             isonuclear master files
c  output: (r*8)  ddens()   = log10(electron density(cm-3)) from adf11
c  output: (r*8)  dtev()    = log10(electron temperature (eV) from adf11
c  output: (r*8)  drcof(,,) = if(iclass <=9):
c                                log10(coll.-rad. coefft.) from
c                                isonuclear master file
c                             if(iclass >=10):
c                                coll.-rad. coefft. from
c                                isonuclear master file
c                             1st dim: index of (sstage, parent, base)
c                                      block in isonuclear master file
c                             2nd dim: electron temperature index
c                             3rd dim: electron density index
c
c  output: (l*4)  lres      = .true.  => partial file
c                           = .false. => not partial file
c  output: (l*4)  lstan     = .true.  => standard file
c                           = .false. => not standard file
c  output: (l*4)  lptn      = .true.  => partition block present
c                           = .false. => partition block not present
c
c  routines:
c          routine    source    brief description
c          ------------------------------------------------------------
c          i4unit     adas      fetch unit number for output of messages
c          i4fctn     adas      convert string to integer form
c          xfelem     adas      return element name given nuclear charge
c          xxword     adas      extract position of number in buffer
c          xxslen     adas      find string less front and tail blanks
c          xxcase     adas      convert a string to upper or lower case
c          xxrptn     adas      analyse an adf11 file partition block
c
c author:  h. p. summers, university of strathclyde
c          ja7.08
c          tel. 0141-548-4196
c
c date:    04/10/06
c
c version: 1.1 date: 04/10/2006
c modified: hugh summers
c - first edition.
c
c version: 1.2 date: 21/01/2007
c modified: Allan Whiteford
c - Commented out warning about lack of iclass,
c                 all of the present ADAS files do not contain
c                 this information
c                 (first commit to CVS)
c
c version: 1.3 date: 08/03/2007
c modified: Hugh Summers
c - adjustments for revised ecd formats.
c                 charge exchange donor/donor mass checks and
c                 dnr_ele, dnr_ams added to parameter return.
c
c-----------------------------------------------------------------------
```

```
CHARACTER*12        DNR_ELE
INTEGER             IBLMX,       ICLASS,      ICNCTV(NDCNCT)
INTEGER             IDDIMD,      IDMAX,       IPTNA(NDPTNL,NDPTN)
INTEGER             IPTNCA(NDPTNL,NDPTN,NDPTNC)
INTEGER             IPTNLA(NDPTNL),           IS1MAX,      IS1MIN
INTEGER             ISDIMD,      ISMAX,       ISPBR(ISDIMD)
INTEGER             ISPPR(ISDIMD),            ISSTGR(ISDIMD)
INTEGER             ITDIMD,      ITMAX,       IUNIT,       IZ0
INTEGER             NCNCT,       NDCNCT,      NDPTN,       NDPTNC
INTEGER             NDPTNL,      NPTN(NDPTNL)
INTEGER             NPTNC(NDPTNL,NDPTN),      NPTNL
LOGICAL             LPTN,        LRES,        LSTAN
REAL*8              DDENS(IDDIMD),            DNR_AMS
REAL*8              DRCOF(ISDIMD,ITDIMD,IDDIMD)
REAL*8              DTEV(ITDIMD)
```

9.100 xxdata_12: Subroutine xxdata_12 from library adaslib

```
      subroutine xxdata_12(iunit  , dsname ,
     &                     ndtem  , ndden  , ndein  , ndzef   , ndmag  ,
     &                     ndstore,
     &                     nbsel  ,
     &                     csymb  , czion  , cwavel , cdonor , crecvr  ,
     &                     ctrans , cfile  , ctype  , cindm  ,
     &                     qefref ,
     &                     enref  , teref  , deref  , zeref   , bmref  ,
     &                     nenera , ntempa , ndensa , nzeffa  , nbmaga ,
     &                     enera  , tempa  , densa  , zeffa   , bmaga  ,
     &                     qenera , qtempa , qdensa , qzeffa  , qbmaga
     &                    )
c-----------------------------------------------------------------------
c
c  ***************** fortran77 subroutine: xxdata_12 ******************
c
c  purpose : To fetch data from input dataset of type adf12.
c
c  notes:    Extensive rewrite of xxdata_12.for which it is a
c            replacement.  Designed to handle automatically produced
c            heavy species data sets.
c
c  calling program: adas316 and general
c
c  subroutine:
c
c  input : (i*4)  iunit     = unit number to read from
c  input : (c*80) dsname    = mvs data set name of data set being read
c  input : (i*4)  ndstore   = maximum number of data blocks allowed
c  input : (i*4)  ndtem     = maximum number of electron temperatures
c  input : (i*4)  ndden     = maximum number of electron densities
c  input : (i*4)  ndein     = maximum number of beam energies
c  input : (i*4)  ndzef     = maximum number of z effectives
c  input : (i*4)  ndmag     = maximum number of magnetic fields
c
c  output: (i*4)  nbsel     = number of blocks present
c
c  output: (c*(*))csymb()   = input data file: element symbol
c                             dimension: data-block index
c  output: (c*(*))czion()   = input data file: emitting ion charge
c                             dimension: data-block index
c  output: (c*(*))cwavel()  = input data file: wavelength (A)
c                             dimension: data-block index
c  output: (c*(*))cdonor()  = input data file: donor neutral atom
c                             dimension: data-block index
c  output: (c*(*))crecvr()  = input data file: receiver nucleus
c                             dimension: data-block index
c  output: (c*(*))ctrans()  = input data file: transition
c                             dimension: data-block index
c  output: (c*(*))cfile()   = input data file: specific ion file source
c                             dimension: data-block index
c  output: (c*(*))ctype()   = input data file: type of emissivity
c                             dimension: data-block index
c  output: (c*(*))cindm()   = file data file: emissivity index
c                             dimension: data-block index
c
c  output: (r*8)  qefref()  = reference value of rate coefficient
c  output: (r*8)  enref()   =     "       "   "  energy
c  output: (r*8)  teref()   =     "       "   "  temperature
```

```
c  output: (r*8)  deref()   =     "       "   "  density
c  output: (r*8)  zeref()   =     "       "   "  effective z
c  output: (r*8)  bmref()   =     "       "   "  magnetic field
c  output: (i*4)  nenera()  = number of energies
c  output: (i*4)  ntempa()  = number of temperatures
c  output: (i*4)  ndensa()  = number of densities
c  output: (i*4)  nzeffa()  = number of effective z's
c  output: (i*4)  nbmaga()  = number of magnetic field values
c                             1st. dim: ndstore
c                             (for above arrays)
c
c  output: (r*8)  enera(,)  = energies
c  output: (r*8)  qenera(,) = rate coefficients for energy value
c  output: (r*8)  tempa(,)  = temperatures
c  output: (r*8)  qtempa(,) = rate coefficients for temperatures
c  output: (r*8)  densa(,)  = densities
c  output: (r*8)  qdensa(,) = rate coefficients for desnities
c  output: (r*8)  zeffa(,)  = effective z
c  output: (r*8)  qzeffa(,) = rate coefficients for effective z
c  output: (r*8)  bmaga(,)  = magnetic field
c  output: (r*8)  qbmaga(,) = rate coefficients for magnetic fields
c                             1st dim: 12 or 24  depending on parameter
c                             2nd dim: ndstore
c
c  routines:
c           routine    source    brief description
c           ------------------------------------------------------------
c           i4unit     adas      fetches file handle for error message
c           xxslen     adas      locate first and last char. of string
c           xxhkey     adas      extract response to key on a text line
c           xxcase     adas      convert a string to upper or lower case
c
c
c  author:  h p summers, university of strathcldye
c  date:    31/05/07
c
c  update:
c
c VERSION  : 1.1
c DATE     : 05-06-2007
c MODIFIED : H P Summers
c              - First version.
c
c-----------------------------------------------------------------------
      CHARACTER*(*)       CDONOR(NDSTORE),          CFILE(NDSTORE)
      CHARACTER*(*)       CINDM(NDSTORE),           CRECVR(NDSTORE)
      CHARACTER*(*)       CSYMB(NDSTORE),           CTRANS(NDSTORE)
      CHARACTER*(*)       CTYPE(NDSTORE),           CWAVEL(NDSTORE)
      CHARACTER*(*)       CZION(NDSTORE),           DSNAME
      INTEGER             IUNIT,       NBMAGA(NDSTORE),          NBSEL
      INTEGER             NDDEN,       NDEIN,       NDENSA(NDSTORE)
      INTEGER             NDMAG,       NDSTORE,     NDTEM,       NDZEF
      INTEGER             NENERA(NDSTORE),          NTEMPA(NDSTORE)
      INTEGER             NZEFFA(NDSTORE)
      REAL*8              BMAGA(NDMAG,NDSTORE),     BMREF(NDSTORE)
      REAL*8              DENSA(NDDEN,NDSTORE),     DEREF(NDSTORE)
      REAL*8              ENERA(NDEIN,NDSTORE),     ENREF(NDSTORE)
      REAL*8              QBMAGA(NDMAG,NDSTORE)
      REAL*8              QDENSA(NDDEN,NDSTORE),    QEFREF(NDSTORE)
      REAL*8              QENERA(NDEIN,NDSTORE)
      REAL*8              QTEMPA(NDTEM,NDSTORE)
```

```
REAL*8              QZEFFA(NDZEF,NDSTORE),     TEMPA(NDTEM,NDSTORE)
REAL*8              TEREF(NDSTORE),            ZEFFA(NDZEF,NDSTORE)
REAL*8              ZEREF(NDSTORE)
```

9.101 xxdata_13: Subroutine xxdata_13 from library adaslib

```
       subroutine xxdata_13( iunit  , dsname ,
     &                       nstore , ntdim  , nddim  ,
     &                       iz0    , iz     , iz1    , esym  ,
     &                       nbsel  , isela  ,
     &                       cwavel , cfile  , ctype  , cindm ,
     &                       ita    , ida    ,
     &                       teta   , teda   ,
     &                       sxb
     &                     )
C-----------------------------------------------------------------------
C
C  ***************** FORTRAN77 SUBROUTINE: xxdata_13 ******************
C
C  PURPOSE:  To fetch data from input ionisations/photon file
C            for a given emitting ion (element and charge).
C
C  CALLING PROGRAM: ADAS503/ssxb
C
C  DATA:     Up to 'nstore' sets (data-blocks) of data may be  read from
C            the file - each block forming a complete set of ionizations
C            per photon values for given temp/density combination. Each
C            data-block  is  analysed independently of any  other  data-
C            block.
C
C            The units used in the data file are taken as follows:
C
C            Temperatures : eV
C            Densities    : cm-3
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  IUNIT    = UNIT TO WHICH INPUT FILE IS ALLOCATED.
C
C          (I*4)  NSTORE   = MAXIMUM NUMBER  OF  INPUT DATA-BLOCKS  THAT
C                            CAN BE STORED.
C          (I*4)  NTDIM    = MAX NUMBER OF ELECTRON TEMPERATURES ALLOWED
C          (I*4)  NDDIM    = MAX NUMBER OF ELECTRON DENSITIES    ALLOWED
C
C  OUTPUT: (I*4)  IZ0      = READ - EMITTING ION - NUCLEAR CHARGE
C          (I*4)  IZ       = READ - EMITTING ION - CHARGE
C          (I*4)  IZ1      = READ - EMITTING ION - CHARGE + 1
C          (C*2)  ESYM     = READ - EMITTING ION - ELEMENT SYMBOL
C
C          (I*4)  NBSEL    = NUMBER OF DATA-BLOCKS ACCEPTED & READ IN.
C          (I*4)  ISELA()  = READ - DATA-SET DATA-BLOCK ENTRY INDICES
C                            DIMENSION: DATA-BLOCK INDEX
C
C          (C*10) CWAVEL() = READ - WAVELENGTH (ANGSTROMS)
C                            DIMENSION: DATA-BLOCK INDEX
C          (C*8)  CFILE()  = READ - SPECIFIC ION FILE SOURCE
C                            DIMENSION: DATA-BLOCK INDEX
C          (C*8)  CTYPE()  = READ - DATA TYPE
C                            DIMENSION: DATA-BLOCK INDEX
C          (C*2)  CINDM()  = READ - METASTABLE INDEX
C                            DIMENSION: DATA-BLOCK INDEX
C
C          (I*4)  ITA()    = READ - NUMBER OF ELECTRON TEMPERATURES
C                            DIMENSION: DATA-BLOCK INDEX
C          (I*4)  IDA()    = READ - NUMBER OF ELECTRON DENSITIES
```

```
C                             DIMENSION: DATA-BLOCK INDEX
C
C          (R*8)  TETA(,)  = READ - ELECTRON TEMPERATURES (UNITS: eV)
C                             1st DIMENSION: ELECTRON TEMPERATURE INDEX
C                             2nd DIMENSION: DATA-BLOCK INDEX
C          (R*8)  TEDA(,)  = READ - ELECTRON DENSITIES (UNITS: CM-3)
C                             1st DIMENSION: ELECTRON DENSITY INDEX
C                             2nd DIMENSION: DATA-BLOCK INDEX
C
C          (R*8)  sxb(,,)   =READ - PHOTON EMISSIVITY VALUES
C                             1st DIMENSION: ELECTRON TEMPERATURE INDEX
C                             2nd DIMENSION: ELECTRON DENSITY INDEX
C                             3rd DIMENSION: DATA-BLOCK INDEX
C
C ROUTINE: (I*4)  I4EIZ0   = FUNCTION - (SEE ROUTINES SECTION BELOW)
C          (I*4)  I4FCTN   = FUNCTION - (SEE ROUTINES SECTION BELOW)
C          (I*4)  I4UNIT   = FUNCTION - (SEE ROUTINES SECTION BELOW)
C          (I*4)  IBLK     = ARRAY INDEX: DATA-BLOCK INDEX
C          (I*4)  ITT      = ARRAY INDEX: ELECTRON TEMPERATURE INDEX
C          (I*4)  ITD      = ARRAY INDEX: ELECTRON DENSITY     INDEX
C          (I*4)  NTNUM    = NUMBER OF ELECTRON TEMPERATURES FOR CURRENT
C                             DATA-BLOCK
C          (I*4)  NDNUM    = NUMBER OF ELECTRON DENSITIES    FOR CURRENT
C                             DATA-BLOCK
C          (I*4)  IABT     = RETURN CODE FROM 'I4FCTN'
C          (I*4)  IPOS1    = GENERAL USE STRING INDEX VARIABLE
C          (I*4)  IPOS2    = GENERAL USE STRING INDEX VARIABLE
C
C          (L*4)  LBEND    = IDENTIFIES WHETHER THE LAST OF THE  INPUT
C                             DATA SUB-BLOCKS HAS BEEN LOCATED.
C                             (.TRUE. => END OF SUB-BLOCKS REACHED)
C
C          (C*1)  CSLASH   = '/' - DELIMITER FOR 'XXHKEY'
C          (C*2)  C2       = GENERAL USE TWO BYTE CHARACTER STRING
C          (C*5)  IONNAM   = EMITTING ION READ FROM DATASET
C          (C*6)  CKEY1    = 'FILMEM' - INPUT BLOCK HEADER KEY
C          (C*4)  CKEY2    = 'TYPE  ' - INPUT BLOCK HEADER KEY
C          (C*4)  CKEY3    = 'INDM  ' - INPUT BLOCK HEADER KEY
C          (C*4)  CKEY4    = 'ISEL  ' - INPUT BLOCK HEADER KEY
C          (C*80) C80      = GENERAL USE 80 BYTE  CHARACTER  STRING  FOR
C                             THE INPUT OF DATA-SET RECORDS.
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          XXHKEY     ADAS      OBTAIN KEY/RESPONSE STRINGS FROM TEXT
C          I4EIZ0     ADAS      INTEGER*4 FUNCTION    -
C                               RETURNS Z0 FOR GIVEN ELEMENT SYMBOL
C          I4FCTN     ADAS      INTEGER*4 FUNCTION    -
C                               CONVERT CHARACTER STRING TO INTEGER
C          I4UNIT     ADAS      INTEGER*4 FUNCTION    -
C                               FETCH UNIT NUMBER FOR OUTPUT OF MESSAGES
C
C
C-----------------------------------------------------------------------
C
C
C NOTES: Based on xxdata_15.for.
C
C
C VERSION  : 1.1
```

```
C DATE     : 17-02-2006
C MODIFIED : Martin O'Mullane
C              - First version
C
C VERSION  : 1.2
C DATE     : 21-01-2007
C MODIFIED : Allan Whiteford
C              - Changed error messages from xxdata_15 to xxdata_13.
C
C-----------------------------------------------------------------------
      CHARACTER*8         CFILE(NSTORE)
      CHARACTER*2         CINDM(NSTORE)
      CHARACTER*8         CTYPE(NSTORE)
      CHARACTER*10        CWAVEL(NSTORE)
      CHARACTER*80        DSNAME
      CHARACTER*2         ESYM
      INTEGER             IDA(NSTORE), ISELA(NSTORE)
      INTEGER             ITA(NSTORE), IUNIT,       IZ,          IZ0
      INTEGER             IZ1,         NBSEL,       NDDIM,       NSTORE
      INTEGER             NTDIM
      REAL*8              SXB(NTDIM,NDDIM,NSTORE),  TEDA(NDDIM,NSTORE)
      REAL*8              TETA(NTDIM,NSTORE)
```

9.102 xxdata_15: Subroutine xxdata_15 from library adaslib

```
       subroutine xxdata_15( iunit  , dsname ,
     &                       nstore , ntdim  , nddim  ,
     &                       ndptnl , ndptn  , ndptnc , ndcnct ,
     &                       ndstack, ndcmt ,
     &                       iz0    , is     , is1    , esym   ,
     &                       nptnl  , nptn   , nptnc  ,
     &                       iptnla , iptna  , iptnca ,
     &                       ncnct  , icnctv ,
     &                       ncptn_stack     , cptn_stack      ,
     &                       lres   , lptn   , lcmt   , lsup   ,
     &                       nbsel  , isela  ,
     &                       cwavel , cfile  , ctype  , cindm  ,
     &                       wavel  , ispbr  , isppr  , isstgr , iszr  ,
     &                       ita    , ida    ,
     &                       teta   , teda   ,
     &                       pec    , pec_max,
     &                       ncmt_stack      , cmt_stack
     &                     )
c-----------------------------------------------------------------------
c
c  ***************** fortran77 subroutine: xxdata_15 ******************
c
c  purpose:  To fetch  data  from  an input photon emissivity file
c            for a given emitting element superstage .
c
c  data:    Up to 'nstore' sets (data-blocks) of data may be  read from
c           the file - each block forming a complete set of photon
c           emissivity coefft. values for given temp/density grid.
c           Each data-block  is  analysed independently of any  other
c           datablock.
c
c           the units used in the data file are taken as follows:
c
c           temperatures : ev
c           densities    : cm-3
c           pec          : phot. cm3 s-1
c
c  subroutine:
c
c  input : (i*4)  iunit    = unit to which input file is allocated.
c          (i*4)  dsname   = name of opened data set on iunit
c
c          (i*4)  nstore   = maximum number  of  input data-blocks  that
c                            can be stored.
c          (i*4)  ntdim    = max number of electron temperatures allowed
c          (i*4)  nddim    = max number of electron densities    allowed
c          (i*4)  ndptnl   = maximum level of partitions
c          (i*4)  ndptn    = maximum no. of partitions in one level
c          (i*4)  ndptnc   = maximum no. of components in a partition
c          (i*4)  ndcnct   = maximum number of elements in connection
c          (i*4)  ndstack  = maximum number of partition text lines
c          (i*4)  ndcmt    = maximum number of comment text lines
c                             vector
c output:  (i*4)  iz0      = read - emitting ion - nuclear charge
c          (i*4)  is       = read - emitting ion - charge
c                            (generalised to superstage label)
c          (i*4)  is1      = read - emitting ion - charge + 1
c                            (generalised to superstage index= is + 1)
c          (c*2)  esym     = read - emitting ion - element symbol
```

```
c
c          (i*4)  nptnl    = number of partition levels in block
c          (i*4)  nptn()   = number of partitions in partition level
c                            1st dim: partition level
c          (i*4)  nptnc(,) = number of components in partition
c                            1st dim: partition level
c                            2nd dim: member partition in partition level
c          (i*4)  iptnla() = partition level label (0=resolved root,1=
c                                                   unresolved root)
c                            1st dim: partition level index
c          (i*4)  iptna(,) = partition member label (labelling starts at 0)
c                            1st dim: partition level index
c                            2nd dim: member partition index in partition
c                            level
c          (i*4)  iptnca(,,)= component label (labelling starts at 0)
c                            1st dim: partition level index
c                            2nd dim: member partition index in partition
c                            level
c                            3rd dim: component index of member partition
c          (i*4)  ncnct     = number of elements in connection vector
c          (i*4)  icnctv()  = connection vector of number of partitions
c                             of each superstage in resolved case
c                             including the bare nucleus
c                             1st dim: connection vector index
c          (i*4)  ncptn_stack = number of text lines in partition block
c          (c*80) cptn_stack()= text lines in partition block
c                               1st dim: text line index (1->ncptn_stack)
c
c          (l*4)  lres      = .true.  => partial file
c                           = .false. => not partial file
c          (l*4)  lptn      = .true.  => partition block present
c                           = .false. => partition block not present
c          (l*4)  lcmt      = .true.  => comment text block present
c                           = .false. => comment text block not present
c          (l*4)  lsup      = .true.  => ss use of filmem field
c                           = .false. => old use of filmem field
c
c          (i*4)  nbsel     = number of data-blocks accepted & read in.
c          (i*4)  isela()   = read - data-set data-block entry indices
c                             dimension: data-block index
c
c          (c*10) cwavel() = wavelength string (angstroms)
c                            1st dim: data-block index
c          (c*8)  cfile()  = specific ion file source string in older
c                            forms.  Field not present in superstage
c                            version, but reused for added information
c                            1st dim: data-block index
c          (c*8)  ctype()  = data type string
c                            1st dim: data-block index
c          (c*2)  cindm()  = metastable index string
c                            1st dim: data-block index
c
c          (r*8)  wavel()   = wavelength (angstroms)
c                             dimension: data-block index
c          (i*4)  isppr()   = parent index for each line block
c                             1st dim: index of block in adf15 file
c          (i*4)  ispbr()   = base index for each line block
c                             1st dim: index of block in adf15 file
c          (i*4)  isstgr()  = s1 for each resolved data block
c                             1st dim: index of block in adf15 file
c          (i*4)  iszr()    = ion charge relating to each line
```

```
c                              1st dim: index of block in adf15 file
c
c          (i*4)  ita()      = number of electron temperatures
c                              dimension: data-block index
c          (i*4)  ida()      = read - number of electron densities
c                              1st dim: data-block index
c
c          (r*8)  teta(,)    =  electron temperatures (units: ev)
c                              1st dim: electron temperature index
c                              2nd dim: data-block index
c          (r*8)  teda(,)    = electron densities (units: cm-3)
c                              1st dim: electron density index
c                              2nd dim: data-block index
c
c          (r*8)  pec(,,)    = photon emissivity coeffts
c                              1st dim: electron temperature index
c                              2nd dim: electron density index
c                              3rd dim: data-block index
c          (r*8)  pec_max()  = photon emissivity coefft. maximum
c                              as a function of Te at first Ne value
c                              1st dim: data-block index
c          (i*4)  ncmt_stack = number of text lines in comment block
c          (c*80) cmt_stack()= text lines in comment block
c                               1st dim: text line index (1->ncmt_stack)
c
c routine: (i*4)  i4eiz0    = function - (see routines section below)
c          (i*4)  i4fctn    = function - (see routines section below)
c          (i*4)  i4unit    = function - (see routines section below)
c          (i*4)  iblk      = array index: data-block index
c          (i*4)  itt       = array index: electron temperature index
c          (i*4)  itd       = array index: electron density     index
c          (i*4)  ntnum     = number of electron temperatures for current
c                             data-block
c          (i*4)  ndnum     = number of electron densities    for current
c                             data-block
c          (i*4)  iabt      = return code from 'i4fctn'
c          (i*4)  ipos1     = general use string index variable
c          (i*4)  ipos2     = general use string index variable
c
c          (l*4)  lbend     = identifies whether the last of the  input
c                             data sub-blocks has been located.
c                             (.true. => end of sub-blocks reached)
c
c          (c*1)  cslash    = '/' - delimiter for 'xxhkey'
c          (c*2)  c2        = general use two byte character string
c          (c*5)  ionnam    = emitting ion read from dataset
c          (c*6)  ckey1     = 'filmem' - input block header key
c          (c*4)  ckey2     = 'type  ' - input block header key
c          (c*4)  ckey3     = 'indm  ' - input block header key
c          (c*4)  ckey4     = 'isel  ' - input block header key
c          (c*80) c80       = general use 80 byte  character  string  for
c                             the input of data-set records.
c
c routines:
c          routine    source    brief description
c          ------------------------------------------------------------
c          i4eiz0     adas      returns z0 for given element symbol
c          i4fctn     adas      convert character string to integer
c          i4unit     adas      fetch unit number for output of messages
c          r8fctn     adas      convert string to real number
c          xxmkrp     adas      make up root partition text lines
```

```
c          xxcase     adas      convert a string to upper or lower case
c          xxhkey     adas      obtain key/response strings from text
c          xxrptn     adas      analyse an adf11 file partition block
c          xxword     adas      extract position of number in buffer
c          xxslen     adas      find string less front and tail blanks
c
c author:  H. P. Summers
c          k1/1/57
c          jet ext. 4941
c
c date:    11/10/91
c
c update:  05/12/91 - PE Briden: ionnam now allowed to occupy either
c                                4 or 5 spaces in the header.
c
c update:  23/04/93 - PE Briden - adas91: added i4unit function to write
c                                         statements for screen messages
c
c update:  24/05/93 - PE Briden - adas91: changed i4unit(0)-> i4unit(-1)
c
c update:  27/2/95  - L. Jalota - idl_adas : increased size dsname for
c                                            use under unix systems
c
c unix-idl port:
c
c version: 1.2                          date: 23-1-96
c modified: Tim Hammond (tessella support services plc)
c               - corrected format statements for dsname length
c
c-----------------------------------------------------------------------
c
c
c notes: copied from e3data.for. this is v1.1 of xxdata_15.
c
c
c version  : 1.1
c date     : 12-04-2005
c modified : Martin o'Mullane
c              - first version
c
c version  : 1.2
c date     : 25-04-2005
c modified : Martin o'Mullane
c              - increase c3 to character*3 to permit more than
c                100 entries in adf15 file.
c
c version  : 1.3
c date     : 15-05-2006
c modified : Hugh Summers
c              - extended to operation with superstages and partitions.
c
c version  : 1.4
c date     : 03-01-2007
c modified : Hugh Summers
c               - remove redundant variables.
c
c-----------------------------------------------------------------------
      CHARACTER*8         CFILE(NSTORE)
      CHARACTER*2         CINDM(NSTORE)
      CHARACTER*80        CMT_STACK(NDCMT),         CPTN_STACK(NDSTACK)
      CHARACTER*8         CTYPE(NSTORE)
```

```
CHARACTER*10        CWAVEL(NSTORE)
CHARACTER*80        DSNAME
CHARACTER*2         ESYM
INTEGER             ICNCTV(NDCNCT),           IDA(NSTORE)
INTEGER             IPTNA(NDPTNL,NDPTN)
INTEGER             IPTNCA(NDPTNL,NDPTN,NDPTNC)
INTEGER             IPTNLA(NDPTNL),           IS,          IS1
INTEGER             ISELA(NSTORE),            ISPBR(NSTORE)
INTEGER             ISPPR(NSTORE),            ISSTGR(NSTORE)
INTEGER             ISZR(NSTORE),             ITA(NSTORE), IUNIT
INTEGER             IZ0,         NBSEL,       NCMT_STACK,  NCNCT
INTEGER             NCPTN_STACK, NDCMT,       NDCNCT,      NDDIM
INTEGER             NDPTN,       NDPTNC,      NDPTNL,      NDSTACK
INTEGER             NPTN(NDPTNL),             NPTNC(NDPTNL,NDPTN)
INTEGER             NPTNL,       NSTORE,      NTDIM
LOGICAL             LCMT,        LPTN,        LRES,        LSUP
REAL*8              PEC(NTDIM,NDDIM,NSTORE),  PEC_MAX(NSTORE)
REAL*8              TEDA(NDDIM,NSTORE),       TETA(NTDIM,NSTORE)
REAL*8              WAVEL(NSTORE)
```

9.103 xxdata_16: Subroutine xxdata_16 from library adaslib

```
       SUBROUTINE xxdata_16( IUNIT  , DSNAME ,
     &                       NSTORE , NTDIM  ,
     &                       ESYM   , IZ0    ,
     &                       NBSEL  , ISELA  ,
     &                       IZ     , IZ1    ,
     &                       CWAVEL , CIION  , CICODE , CISCRP , CITYPE ,
     &                       ITA    ,
     &                       TMA    , TETA   , DENSA  , GCF
     &                     )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: E7DATA *********************
C
C  PURPOSE:  TO  FETCH  DATA  FROM  INPUT CONTRIBUTION FUNCTIONS
C             OF AN ELEMENT AND ITS IONS.
C            (MEMBER STORED IN IONELEC.DATA - MEMBER PREFIX 'GCF#').
C
C  CALLING PROGRAM: ADAS507/SGCF
C
C  DATA:
C
C           UP TO 'NSTORE' SETS (DATA-BLOCKS) OF DATA MAY BE  READ FROM
C           THE FILE - EACH BLOCK FORMING A COMPLETE SET OF CONTRIBUTION
C           FUNCTION VALUES FOR GIVEN TEMPERATURES.
C           EACH DATA-BLOCK  IS  ANALYSED INDEPENDENTLY OF ANY  OTHER
C           DATA-BLOCK.
C
C           THE UNITS USED IN THE DATA FILE ARE TAKEN AS FOLLOWS:
C
C           MODEL PARAMETER     : UNSPECIFIED
C           DENSITIES           : CM-3
C           TEMPERATURES        : EV
C           CONTR. FUNCTION     : CM**3 S-1
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  IUNIT    = UNIT TO WHICH INPUT FILE IS ALLOCATED.
C  INPUT : (C*44) DSNAME   = MVS DATA SET NAME OF DATA SET BEING READ
C
C  INPUT : (I*4)  NSTORE   = MAXIMUM NUMBER  OF  INPUT DATA-BLOCKS  THAT
C                            CAN BE STORED.
C  INPUT : (I*4)  NTDIM    = MAX NUMBER OF ELECTRON TEMPERATURES ALLOWED
C
C  OUTPUT: (C*2)  ESYM     = READ - RADIATING ION - ELEMENT SYMBOL
C  OUTPUT: (I*4)  IZ0      = READ - RADIATING ION - NUCLEAR CHARGE
C
C  OUTPUT: (I*4)  NBSEL    = NUMBER OF DATA-BLOCKS ACCEPTED & READ IN.
C  OUTPUT: (I*4)  ISELA()  = READ - DATA-SET DATA-BLOCK ENTRY INDICES
C                            DIMENSION: DATA-BLOCK INDEX
C
C  OUTPUT: (I*4)  IZ()     = READ - RADIATING ION CHARGE
C                                   ( SET TO -1 IF WHOLE ELEMENT)
C                            DIMENSION: DATA-BLOCK INDEX
C  OUTPUT: (I*4)  IZ1()    = READ - RADIATING ION CHARGE +1
C                                   ( SET TO 1 IF WHOLE ELEMENT)
C                            DIMENSION: DATA-BLOCK INDEX
C
C  OUTPUT: (C*10) CWAVEL() = READ - WAVELENGTH (ANGSTROMS)
C                            DIMENSION: DATA-BLOCK INDEX
```

```
C  OUTPUT: (C*5)  CIION()  = READ - RADIATING ION (AS <ESYM>+(IZ()> )
C                           DIMENSION: DATA-BLOCK INDEX
C  OUTPUT: (C*8)  CICODE() = READ - SOURCE PROGRAM
C                           DIMENSION: DATA-BLOCK INDEX
C  OUTPUT: (C*8)  CISCRP() = READ - SOURCE SCRIPT
C                           DIMENSION: DATA-BLOCK INDEX
C  OUTPUT: (C*5)  CITYPE() = READ - RADIATION TYPE
C                           DIMENSION: DATA-BLOCK INDEX
C
C  OUTPUT: (I*4)  ITA()    = READ - NUMBER OF ELECTRON TEMPERATURES
C                           DIMENSION: DATA-BLOCK INDEX
C
C  OUTPUT: (R*8)  TMA(,)   = READ - MODEL PARAMETER (UNITS: UNDEFINED)
C                           1st DIMENSION: ELECTRON TEMPERATURE INDEX
C                           2nd DIMENSION: DATA-BLOCK INDEX
C
C  OUTPUT: (R*8)  TETA(,)  = READ - ELECTRON TEMPERATURES (UNITS: EV)
C                           1st DIMENSION: ELECTRON TEMPERATURE INDEX
C                           2nd DIMENSION: DATA-BLOCK INDEX
C
C  OUTPUT: (R*8)  DENSA(,) = READ - ELECTRON DENSITIES    (UNITS: CM-3)
C                           1st DIMENSION: ELECTRON TEMPERATURE INDEX
C                           2nd DIMENSION: DATA-BLOCK INDEX
C
C  OUTPUT: (R*8)  GCF(,)    =READ - FULL SET OF GENERALISED CONTIBUTION
C                                   FUNCTIONS (CM**3 S-1)
C                           1st DIMENSION: ELECTRON TEMPERATURE INDEX
C                           2nd DIMENSION: DATA-BLOCK INDEX
C
C          (I*4)  I4EIZ0   = FUNCTION - (SEE ROUTINES SECTION BELOW)
C          (I*4)  I4FCTN   = FUNCTION - (SEE ROUTINES SECTION BELOW)
C          (I*4)  I4UNIT   = FUNCTION - (SEE ROUTINES SECTION BELOW)
C          (I*4)  IBLK     = ARRAY INDEX: DATA-BLOCK INDEX
C          (I*4)  ITT      = ARRAY INDEX: ELECTRON TEMPERATURE INDEX
C          (I*4)  NTNUM    = NUMBER OF ELECTRON TEMPERATURES FOR CURRENT
C                           DATA-BLOCK
C          (I*4)  IABT     = RETURN CODE FROM 'I4FCTN'
C          (I*4)  IPOS1    = GENERAL USE STRING INDEX VARIABLE
C          (I*4)  IPOS2    = GENERAL USE STRING INDEX VARIABLE
C
C          (R*8)  R8FCTN   = FUNCTION - (SEE ROUTINES SECTION BELOW)
C
C          (L*4)  LBEND    = IDENTIFIES WHETHER THE LAST OF THE  INPUT
C                           DATA SUB-BLOCKS HAS BEEN LOCATED.
C                           (.TRUE. => END OF SUB-BLOCKS REACHED)
C
C          (C*1)  CSLASH   = '/' - DELIMITER FOR 'XXHKEY'
C          (C*2)  C2       = GENERAL USE TWO BYTE CHARACTER STRING
C          (C*4)  CKEY1    = 'CODE'  - INPUT BLOCK HEADER KEY
C          (C*6)  CKEY2    = 'SCRIPT - INPUT BLOCK HEADER KEY
C          (C*4)  CKEY3    = 'TYPE'  - INPUT BLOCK HEADER KEY
C          (C*4)  CKEY4    = 'ISEL'  - INPUT BLOCK HEADER KEY
C          (C*10) C10      = GENERAL USE TEN BYTE CHARACTER STRING
C          (C*80) C80      = GENERAL USE 80 BYTE  CHARACTER  STRING  FOR
C                           THE INPUT OF DATA-SET RECORDS.
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          XXHKEY     ADAS      OBTAIN KEY/RESPONSE STRINGS FROM TEXT
C          I4EIZ0     ADAS      INTEGER*4 FUNCTION    -
```

```
C                               RETURNS Z0 FOR GIVEN ELEMENT SYMBOL
C          I4FCTN     ADAS      INTEGER*4 FUNCTION    -
C                               CONVERT CHARACTER STRING TO INTEGER
C          I4UNIT     ADAS      INTEGER*4 FUNCTION    -
C                               FETCH UNIT NUMBER FOR OUTPUT OF MESSAGES
C          R8FCTN     ADAS      REAL*8 FUNCTION       -
C                               CONVERT CHARACTER STRING TO REAL*8
C
C AUTHOR:  H. P. SUMMERS, JET
C          K1/1/57
C          JET EXT. 4941
C
C DATE:    15/04/94
C
C UPDATE:
C
C-----------------------------------------------------------------------
      CHARACTER*8         CICODE(NSTORE)
      CHARACTER*5         CIION(NSTORE)
      CHARACTER*8         CISCRP(NSTORE)
      CHARACTER*5         CITYPE(NSTORE)
      CHARACTER*10        CWAVEL(NSTORE)
      CHARACTER*80        DSNAME
      CHARACTER*2         ESYM
      INTEGER             ISELA(NSTORE),            ITA(NSTORE), IUNIT
      INTEGER             IZ(NSTORE),  IZ0,         IZ1(NSTORE), NBSEL
      INTEGER             NSTORE,      NTDIM
      REAL*8              DENSA(NTDIM,NSTORE),      GCF(NTDIM,NSTORE)
      REAL*8              TETA(NTDIM,NSTORE),       TMA(NTDIM,NSTORE)
```

9.104 xxdata_19: Subroutine xxdata_19 from library adaslib

```
       SUBROUTINE XXDATA_19( IUNIT  , DSNAME ,
     &                       NSTORE , NTDIM  ,
     &                       ESYM   , IZ0    ,
     &                       NBSEL  , ISELA  ,
     &                       IZ     , IZ1    ,
     &                       CIION  , CITYPE , CIINFO ,
     &                       ITA    ,
     &                       TETA   , PZD
     &                     )
C-----------------------------------------------------------------------
C
C  ***************** FORTRAN77 SUBROUTINE: XXDATA_19 ******************
C
C  PURPOSE:  TO  FETCH  DATA  FROM  INPUT RADIATED POWER COEFFICIENTS
C             OF AN ELEMENT AND ITS IONS.
C            (MEMBER STORED IN IONELEC.DATA - MEMBER PREFIX 'PZD#').
C
C  CALLING PROGRAM: ADAS504/SPZD
C
C  DATA:
C
C           UP TO 'NSTORE' SETS (DATA-BLOCKS) OF DATA MAY BE  READ FROM
C           THE FILE - EACH BLOCK FORMING A COMPLETE SET OF  RADIATED
C           POWER COEFFICIENT VALUES FOR GIVEN TEMPERATURES.
C           EACH DATA-BLOCK  IS  ANALYSED INDEPENDENTLY OF ANY  OTHER
C           DATA-BLOCK.
C
C           THE UNITS USED IN THE DATA FILE ARE TAKEN AS FOLLOWS:
C
C           TEMPERATURES        : EV
C           RATE COEFFT         : W CM**3
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  IUNIT    = UNIT TO WHICH INPUT FILE IS ALLOCATED.
C  INPUT : (C*80) DSNAME   = NAME OF DATA SET BEING READ
C
C  INPUT : (I*4)  NSTORE   = MAXIMUM NUMBER  OF  INPUT DATA-BLOCKS  THAT
C                            CAN BE STORED.
C  INPUT : (I*4)  NTDIM    = MAX NUMBER OF ELECTRON TEMPERATURES ALLOWED
C
C  OUTPUT: (C*2)  ESYM     = READ - IONISING ION - ELEMENT SYMBOL
C  OUTPUT: (I*4)  IZ0      = READ - IONISING ION - NUCLEAR CHARGE
C
C  OUTPUT: (I*4)  NBSEL    = NUMBER OF DATA-BLOCKS ACCEPTED & READ IN.
C  OUTPUT: (I*4)  ISELA()  = READ - DATA-SET DATA-BLOCK ENTRY INDICES
C                            DIMENSION: DATA-BLOCK INDEX
C
C  OUTPUT: (I*4)  IZ()     = READ - RADIATING ION CHARGE
C                                   ( SET TO -1 IF WHOLE ELEMENT)
C                            DIMENSION: DATA-BLOCK INDEX
C  OUTPUT: (I*4)  IZ1()    = READ - RADIATING ION CHARGE +1
C                                   ( SET TO 1 IF WHOLE ELEMENT)
C                            DIMENSION: DATA-BLOCK INDEX
C
C  OUTPUT: (C*5)  CIION()  = READ - RADIATING ION (AS <ESYM>+(IZ()> )
C                            DIMENSION: DATA-BLOCK INDEX
C  OUTPUT: (C*5)  CITYPE() = READ - RADIATION TYPE
C                            DIMENSION: DATA-BLOCK INDEX
```

```
C  OUTPUT: (C*20) CIINFO() = READ - INFORMATION STRING
C                            DIMENSION: DATA-BLOCK INDEX
C
C  OUTPUT: (I*4)  ITA()    = READ - NUMBER OF ELECTRON TEMPERATURES
C                            DIMENSION: DATA-BLOCK INDEX
C
C  OUTPUT: (R*8)  TETA(,)  = READ - ELECTRON TEMPERATURES (UNITS: eV)
C                            1st DIMENSION: ELECTRON TEMPERATURE INDEX
C                            2nd DIMENSION: DATA-BLOCK INDEX
C
C  OUTPUT: (R*8)  PZD(,)    =READ - FULL SET OF ZERO DENSITY RADIATED
C                                   POWER COEFFTS. (W CM**3)
C                            1st DIMENSION: ELECTRON TEMPERATURE INDEX
C                            2nd DIMENSION: DATA-BLOCK INDEX
C
C          (I*4)  I4EIZ0   = FUNCTION - (SEE ROUTINES SECTION BELOW)
C          (I*4)  I4FCTN   = FUNCTION - (SEE ROUTINES SECTION BELOW)
C          (I*4)  I4UNIT   = FUNCTION - (SEE ROUTINES SECTION BELOW)
C          (I*4)  IBLK     = ARRAY INDEX: DATA-BLOCK INDEX
C          (I*4)  ITT      = ARRAY INDEX: ELECTRON TEMPERATURE INDEX
C          (I*4)  NTNUM    = NUMBER OF ELECTRON TEMPERATURES FOR CURRENT
C                            DATA-BLOCK
C          (I*4)  IABT     = RETURN CODE FROM 'I4FCTN'
C          (I*4)  IPOS1    = GENERAL USE STRING INDEX VARIABLE
C          (I*4)  IPOS2    = GENERAL USE STRING INDEX VARIABLE
C
C
C          (L*4)  LBEND    = IDENTIFIES WHETHER THE LAST OF THE  INPUT
C                            DATA SUB-BLOCKS HAS BEEN LOCATED.
C                            (.TRUE. => END OF SUB-BLOCKS REACHED)
C
C          (C*1)  CSLASH   = '/' - DELIMITER FOR 'XXHKEY'
C          (C*2)  C2       = GENERAL USE TWO BYTE CHARACTER STRING
C          (C*4)  CKEY1    = '****'  - INPUT BLOCK HEADER KEY
C          (C*5)  CKEY2    = 'TYPE'  - INPUT BLOCK HEADER KEY
C          (C*5)  CKEY3    = 'INFO'  - INPUT BLOCK HEADER KEY
C          (C*4)  CKEY4    = 'ISEL'  - INPUT BLOCK HEADER KEY
C          (C*10) C10      = GENERAL USE TEN BYTE CHARACTER STRING
C          (C*80) C80      = GENERAL USE 80 BYTE  CHARACTER  STRING  FOR
C                            THE INPUT OF DATA-SET RECORDS.
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          XXHKEY     ADAS      OBTAIN KEY/RESPONSE STRINGS FROM TEXT
C          I4EIZ0     ADAS      INTEGER*4 FUNCTION    -
C                               RETURNS Z0 FOR GIVEN ELEMENT SYMBOL
C          I4FCTN     ADAS      INTEGER*4 FUNCTION    -
C                               CONVERT CHARACTER STRING TO INTEGER
C          I4UNIT     ADAS      INTEGER*4 FUNCTION    -
C                               FETCH UNIT NUMBER FOR OUTPUT OF MESSAGES
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/37
C          JET EXT. 2520
C
C DATE:    07/06/91
C
C UPDATE:  23/04/93 - PE BRIDEN - ADAS91: ADDED I4UNIT FUNCTION TO WRITE
C                                         STATEMENTS FOR SCREEN MESSAGES
C
```

```
C UPDATE:  24/05/93 - PE BRIDEN - ADAS91: CHANGED I4UNIT(0)-> I4UNIT(-1)
C
C UPDATE:   4/11/94 - L. JALOTA - INCREASED SIZE OF DSNAME TO RUN UNDER
C                                 UNIX
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 17-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST VERSION
C
C VERSION: 1.2                          DATE: 17-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)/HUGH SUMMERS
C               - MINOR MODIFICATIONS TO FORMAT STATEMENTS
C
C-----------------------------------------------------------------------
C
C NOTES: Copied from e4data.for. This is v1.1 of xxdata_19.
C
C VERSION  : 1.1
C DATE     : 03-07-2008
C MODIFIED : Martin O'Mullane
C              - First version
C
C-----------------------------------------------------------------------
      CHARACTER*20        CIINFO(NSTORE)
      CHARACTER*5         CIION(NSTORE),            CITYPE(NSTORE)
      CHARACTER*80        DSNAME
      CHARACTER*2         ESYM
      INTEGER             ISELA(NSTORE),            ITA(NSTORE), IUNIT
      INTEGER             IZ(NSTORE),  IZ0,         IZ1(NSTORE), NBSEL
      INTEGER             NSTORE,      NTDIM
      REAL*8              PZD(NTDIM,NSTORE),        TETA(NTDIM,NSTORE)
```

9.105 xxdata_20: Subroutine xxdata_20 from library adaslib

```
       SUBROUTINE xxdata_20( IUNIT  , NDLEV  , NDTEM , NDTRN ,
     &                       ELEM   , IZ     , IZ0   , IZ1   ,
     &                       IL     ,
     &                       IA     , CSTRGA , ISA   , ILA   , XJA    ,
     &                       NV     ,
     &                       TEA    , DENSA  , PRESA , RNHNE , TMA    ,
     &                       ITRAN  ,
     &                       I1A    , I2A    , APWL  , SWL   , GFT    ,
     &                       LVALID , INDX
     &                     )
C-----------------------------------------------------------------------
C
C  ***************** FORTRAN77 SUBROUTINE: xxdata_20 ******************
C
C  PURPOSE:  TO FETCH DATA FROM INPUT GFT DATA SET OF TYPE ADF20.
C
C  CALLING PROGRAM: ADAS506
C
C           THE UNITS USED IN THE DATA FILE ARE TAKEN AS FOLLOWS:
C
C           ELECTRON TEMPERATURE: KELVIN
C           ELECTRON DENSITY    : CM-3
C           ELECTRON PRESSURE   : K CM-3
C           TIME                : NOT SPECIFIED
C           WAVELENGTH          : ANGSTROM
C           GFT COEFFT.         : CM3 SEC-1
C
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  IUNIT   = UNIT TO WHICH INPUT FILE IS ALLOCATED
C  INPUT : (I*4)  NDLEV   = MAXIMUM NUMBER OF LEVELS THAT CAN BE READ
C  INPUT : (I*4)  NDTEM   = MAXIMUM NUMBER OF TEMPERATURES
C  INPUT : (I*4)  NDTRN   = MAX. NUMBER OF TRANSITIONS THAT CAN BE READ
C
C  OUTPUT: (C*2)  ELEM    = ELEMENT SYMBOL.
C  OUTPUT: (I*4)  IZ      =  RECOMBINED ION CHARGE READ
C  OUTPUT: (I*4)  IZ0     =         NUCLEAR CHARGE READ
C  OUTPUT: (I*4)  IZ1     = RECOMBINING ION CHARGE READ
C                           (NOTE: IZ1 SHOULD EQUAL IZ+1)
C
C  OUTPUT: (I*4)  IL      = INPUT DATA FILE: NUMBER OF ENERGY LEVELS
C
C  OUTPUT: (I*4)  IA()    = ENERGY LEVEL INDEX NUMBER
C  OUTPUT: (C*18) CSTRGA()= NOMENCLATURE/CONFIGURATION FOR LEVEL 'IA()'
C  OUTPUT: (I*4)  ISA()   = MULTIPLICITY FOR LEVEL 'IA()'
C                           NOTE: (ISA-1)/2 = QUANTUM NUMBER (S)
C  OUTPUT: (I*4)  ILA()   = QUANTUM NUMBER (L) FOR LEVEL 'IA()'
C  OUTPUT: (R*8)  XJA()   = QUANTUM NUMBER (J-VALUE) FOR LEVEL 'IA()'
C                           NOTE: (2*XJA)+1 = STATISTICAL WEIGHT
C
C  OUTPUT: (I*4)  NV      = INPUT DATA FILE: NUMBER OF TEMP/DENS/PRESS/
C                           TIME SETS
C  OUTPUT: (R*8)  TEA()   = INPUT DATA FILE: ELECTRON TEMPERATURES (K)
C  OUTPUT: (R*8)  DENSA() = INPUT DATA FILE: ELECTRON DENSITIES (CM-3)
C  OUTPUT: (R*8)  PRESA() = INPUT DATA FILE: ELECTRON PRESSURES (K CM-3)
C  OUTPUT: (R*8)  TMA()   = INPUT DATA FILE: TIMES OR ARBITRARY (S ?)
C
C  OUTPUT: (I*4)  ITRAN   = INPUT DATA FILE: NUMBER OF TRANSITIONS
```

```
C
C  OUTPUT: (I*4)  I1A()   = TRANSITION:
C                           LOWER ENERGY LEVEL INDEX
C  OUTPUT: (I*4)  I2A()   = TRANSITION:
C                           UPPER ENERGY LEVEL INDEX
C  OUTPUT: (R*8)  APWL()  = APPROXIMATE TRANSITION WAVELENGTH (A)
C  OUTPUT: (R*8)  SWL()   = EXACT TRANSITION WAVELENGTH (A)  (IF SET)
C  OUTPUT: (R*8)  GFT(,)  = GFT COEFFICIENT FOR TRANSITION (CM3 S-1)
C                           1ST DIMENSION - TEMPERATURE 'TEA()'
C                           2ND DIMENSION - TRANSITION INDEX
C  OUTPUT: (L*4)  LVALID  = .TRUE. DATA SET READ AND APPEARS VALID
C                         = .FALSE. ERROR DETECTED IN READING DATA SET
C
C
C          (I*4)  I4UNIT  = FUNCTION (SEE ROUTINE SECTION BELOW)
C          (I*4)  I4FCTN  = FUNCTION (SEE ROUTINE SECTION BELOW)
C          (I*4)  I4EIZ0  = FUNCTION (SEE ROUTINE SECTION BELOW)
C          (I*4)  I       = GENERAL USE.
C          (I*4)  IABT    = RETURN CODE FROM 'I4FCTN'
C          (I*4)  J       = GENERAL USE.
C          (I*4)  K       = GENERAL USE.
C          (I*4)  NVAL    = GENERAL USE
C          (I*4)  IRECL   = RECORD LENGTH OF INPUT DATASET (<=128)
C
C
C          (C*1)  CSLASH  = '/' - DELIMITER FOR 'XXHKEY'
C          (C*4)  C4      = GENERAL USE FOUR BYTE CHARACTER
C          (C*5)  IONNAM  = EMITTING ION READ FROM DATA SET
C          (C*7)  CKEY1   = 'NLEVELS' - INPUT HEADER KEY
C          (C*6)  CKEY2   = 'NKNOTS'  - INPUT HEADER KEY
C          (C*6)  CKEY3   = 'NLINES'  - INPUT HEADER KEY
C          (C*3)  TITLED  = ELEMENT SYMBOL INCLUDING '+'
C          (C*80) CLINE   = CURRENT ENERGY LEVEL INDEX PARAMETER LINE
C          (C*127)BUFFER  = GENERAL STRING BUFFER STORAGE
C
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          XXWORD     ADAS      EXTRACT POSITION OF NUMBER IN BUFFER
C          I4UNIT     ADAS      FETCH UNIT NUMBER FOR OUTPUT OF MESSAGES
C          R8FCTN     ADAS      CONVERTS FROM CHARACTER TO REAL VARIABLE
C          I4FCTN     ADAS      CONVERTS CHARACTER STRING TO INTEGER
C          I4EIZ0     ADAS      RETURNS Z0 FOR GIVEN ELEMENT SYMBOL
C
C
C ROUTINES: NONE
C
C AUTHOR:  H. P. SUMMERS, JET
C          K1/1/57
C          JET EXT. 4941
C
C DATE:    07/04/94
C
C UPDATE:  APR18-95
C          A. C. LANZAFAME, DPAP UNIVERSITY OF STRATHCLYDE
C          TRANSITION INDEX (INDX) ADDED. USED IN DEM CODES
C          TO IDENTIFY THE TRANSITION
C
C          CHARACTER CSTRGA(NDLEV)*(*) changed to CHARACTER*18 CSTRGA(NDLEV)
C          after experinced unstable behaviour on Sun workstation
```

```
C
C UPDATE:
C VERSION:      1.2             DATE:   09-11-95
C MODIFIED: Alessandro Lanzafame
C           - Commented out superfluous variables
C
C-----------------------------------------------------------------------
C
C NOTES: Copied from e6data.for. This is v1.1 of xxdata_20.
C
C
C VERSION  : 1.1
C DATE     : 06-06-2003
C MODIFIED : Martin O'Mullane
C              - First version
C
C-----------------------------------------------------------------------
      CHARACTER*18        CSTRGA(NDLEV)
      CHARACTER*2         ELEM
      INTEGER             I1A(NDTRN),  I2A(NDTRN),  IA(NDLEV),   IL
      INTEGER             ILA(NDLEV),  INDX(NDTRN), ISA(NDLEV),  ITRAN
      INTEGER             IUNIT,       IZ,          IZ0,         IZ1
      INTEGER             NDLEV,       NDTEM,       NDTRN,       NV
      LOGICAL             LVALID
      REAL*8              APWL(NDTRN), DENSA(NDTEM)
      REAL*8              GFT(NDTEM,NDTRN),         PRESA(NDTEM)
      REAL*8              RNHNE(NDTEM),             SWL(NDTRN)
      REAL*8              TEA(NDTEM),  TMA(NDTEM),  XJA(NDLEV)
```

9.106 xxdata_21: Subroutine xxdata_21 from library adaslib

```
C
      SUBROUTINE xxdata_21( IUNIT  , MXBE   , MXTD   , MXTT   ,
     &                      ITZ    , TSYM   , BEREF  , TDREF  ,
     &                      TTREF  , SVREF  , NBE    , BE     ,
     &                      NTDENS , TDENS  , NTTEMP , TTEMP  ,
     &                      SVT    , SVED   , DSNIN
     &                    )


C-----------------------------------------------------------------------
C
C  ***************** FORTRAN77 SUBROUTINE: xxdata_21 ******************
C
C  PURPOSE: TO READ DATA FROM AN EFFECTIVE BEAM STOPPING DATA SET.
C           (ADAS FORMAT ADF21).
C
C  CALLING PROGRAM: SBMS / ADAS304
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  IUNIT     = UNIT TO WHICH DATA SET IS CONNECTED.
C  INPUT : (I*4)  MXBE      = MAXIMUM NUMBER OF BEAM ENERGIES WHICH CAN
C                             BE READ.
C  INPUT : (I*4)  MXTD      = MAXIMUM NUMBER OF TARGET DENSITIES WHICH
C                             CAN BE READ.
C  INPUT : (I*4)  MXTT      = MAXIMUM NUMBER OF TARGET TEMPERATURES
C                             WHICH CAN BE READ.
C  INPUT : (C*80) DSNIN     = NAME OF FILE TO BE READ.
C  OUTPUT: (I*4)  ITZ       = TARGET ION CHARGE.
C  OUTPUT: (C*2)  TSYM      = TARGET ION ELEMENT SYMBOL.
C  OUTPUT: (R*8)  BEREF     = REFERENCE BEAM ENERGY.
C                             UNITS: EV/AMU
C  OUTPUT: (R*8)  TDREF     = REFERENCE TARGET DENSITY.
C                             UNITS: CM-3
C  OUTPUT: (R*8)  TTREF     = REFERENCE TARGET TEMPERATURE.
C                             UNITS: EV
C  OUTPUT: (R*8)  SVREF     = STOPPING COEFFT. AT REFERENCE BEAM ENERGY,
C                             TARGET DENSITY AND TEMPERATURE.
C                             UNITS: CM3 S-1
C  OUTPUT: (I*4)  NBE       = NUMBER OF BEAM ENERGIES.
C  OUTPUT: (R*8)  BE()      = BEAM ENERGIES.
C                             UNITS: EV/AMU
C                             DIMENSION: MXBE
C  OUTPUT: (I*4)  NTDENS    = NUMBER OF TARGET DENSITIES.
C  OUTPUT: (R*8)  TDENS()   = TARGET DENSITIES.
C                             UNITS: CM-3
C                             DIMENSION: MXTD
C  OUTPUT: (I*4)  NTTEMP    = NUMBER OF TARGET TEMPERATURES.
C  OUTPUT: (R*8)  TTEMP()   = TARGET TEMPERATURES.
C                             UNITS: EV
C                             DIMENSION: MXTT
C  OUTPUT: (R*8)  SVT()     = STOPPING COEFFT. AT REFERENCE BEAM ENERGY
C                             AND TARGET DENSITY.
C                             UNITS: CM3 S-1
C                             DIMENSION: MXTT
C  OUTPUT: (R*8)  SVED(,)   = STOPPING COEFFT. AT REFERENCE TARGET
C                             TEMPERATURE.
C                             UNITS: CM3 S-1
C                             1ST DIMENSION: MXBE
```

```
C                             2ND DIMENSION: MXTD
C
C          (I*4)  I          = ARRAY / LOOP INDEX.
C          (I*4)  J          = ARRAY INDEX.
C
C          (C*80) LINE       = TEXT LINE IN DATA SET.
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4UNIT     ADAS      RETURNS UNIT NO. FOR OUTPUT OF MESSAGES.
C
C AUTHOR:  JONATHAN NASH  (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/87
C          JET EXT. 5183
C
C DATE:    07/12/93
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 16-11-95
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST VERSION
C
C-----------------------------------------------------------------------
C
C NOTES: Copied from c4data.for. This is v1.1 of xxdata_21.
C
C
C VERSION  : 1.1
C DATE     : 06-02-2004
C MODIFIED : Martin O'Mullane
C              - First version
C              - File unit is closed within the subroutine.
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      CHARACTER*80        DSNIN
      CHARACTER*2         TSYM
      INTEGER             ITZ,          IUNIT,       MXBE,         MXTD
      INTEGER             MXTT,         NBE,         NTDENS,       NTTEMP
      REAL*8              BE(MXBE),     BEREF,       SVED(MXBE,MXTD)
      REAL*8              SVREF,        SVT(MXTT),   TDENS(MXTD), TDREF
      REAL*8              TTEMP(MXTT), TTREF
```

9.107 xxdata_23: Subroutine xxdata_23 from library adaslib

```
      subroutine xxdata_23(iunit    ,
     &                     ndlev    , ndmet     , ndtem     , ndtext   ,
     &                     seq      , iz0       , iz        , iz1      ,
     &                     ctype    ,
     &                     bwno_f   , nlvl_f    , lmet_f    , lcstrg_f ,
     &                     ia_f     , code_f    , cstrga_f  ,
     &                     isa_f    , ila_f     , xja_f     , wa_f     ,
     &                     nmet_f   , imeta_f   ,
     &                     bwno_i   , nlvl_i    , lmet_i    , lcstrg_i ,
     &                     ia_i     , code_i    , cstrga_i  ,
     &                     isa_i    , ila_i     , xja_i     , wa_i     ,
     &                     nmet_i   , imeta_i   ,
     &                     nte_ion  , tea_ion   , lqred_ion , qred_ion ,
     &                     nf_a     , indf_a    , lyld_a    , yld_a    ,
     &                     nte_exc  , tea_exc   , lqred_exc , qred_exc ,
     &                     l_ion    , l_aug     , l_exc     ,
     &                     ntext    , ctext
     &                    )

c-----------------------------------------------------------------------
c  ****************** fortran77 subroutine: xxdata_23 ******************
c
c  purpose:  to fetch data from an adf23 data set.
c
c  input : (i*4)  iunit       = unit to which input file is allocated
c          (i*4)  ndlev       = maximum number of energy levels in
c                               either ion stage
c          (i*4)  ndmet       = maximum number of metastables
c          (i*4)  ndtem       = maximum number of temperatures
c          (i*4)  ndtext      = maximum number of comment text lines
c
c  output: (c*2)  seq         = iso-electronic sequence symbol
c          (i*4)  iz0         = nuclear charge
c          (i*4)  iz          = ionising ion charge
c          (i*4)  iz1         = ionised ion charge (=iz+1)
c          (c*2)  ctype       = adf23 file resol. ('ca', 'ls' or 'ic')
c          (r*8)  bwno_f      = ionis. poten. of ionised ion (cm-1)
c          (i*4)  nlvl_f      = number of levels of ionised ion
c          (l*4)  lmet_f      = .true.  => ionised metastables marked
c                               .false. => ionised metastables unmarked
c                                          (default action - mark ground)
c          (l*4)  lcstrg_f    = .true.  => standard config strings for
c                                          ionised ion states
c                               .false. => unreadable config string for
c                                          at least one ionised ion state
c          (i*4)  ia_f()      = index of ionised ion levels
c                               1st dim: ionised ion level index
c          (c*1)  code_f()    = met. or excit. DR parent marker (* or #)
c                               1st dim: ionised ion level index
c          (i*(*))cstrga_f()  = ionised ion configuration strings
c                               1st dim: ionised ion level index
c          (i*4)  isa_f()     = ionised ion level multiplicity
c                               1st dim: ionised ion level index
c          (i*4)  ila_f()     = ionised ion total orb. ang. mom.
c                               1st dim: ionised ion level index
c          (r*8)  xja_f()     = ionised ion level (stat wt-1)/2
c                               1st dim: ionised ion level index
c          (r*8)  wa_f()      = ionised ion level wave number (cm-1)
c                               1st dim: ionised ion level index
```

```
c          (i*4)  nmet_f      = number of ionised ion metastables
c          (i*4)  imeta_f()   = pointers to ionised metastables in full
c                               ionised ion state list
c                               1st dim: ionised metastable index
c          (r*8)  bwno_i      = ionis. poten. of ionising ion (cm-1)
c          (i*4)  nlvl_i      = number of levels of ionising ion
c          (l*4)  lmet_i      = .true.  => ionising metastable marked
c                               .false. => ionising metastables unmarked
c                                          (default action - mark ground)
c          (l*4)  lcstrg_i    = .true.  => standard config strings for
c                                          ionising ion states
c                               .false. => unreadable config string for
c                                          at least one ionising ion state
c          (i*4)  ia_i()      = index of ionising ion levels
c                               1st dim: ionising ion level index
c          (c*1)  code_i()    = met. or excit. DR parent marker (* or #)
c                               1st dim: ionising ion level index
c          (i*(*))cstrga_i()  = ionising ion configuration strings
c                               1st dim: ionising ion level index
c          (i*4)  isa_i()     = ionising ion level multiplicity
c                               1st dim: ionising ion level index
c          (i*4)  ila_i()     = ionising ion total orb. ang. mom.
c                               1st dim: ionising ion level index
c          (r*8)  xja_i()     = ionising ion level (stat wt-1)/2
c                               1st dim: ionising ion level index
c          (r*8)  wa_i()      = ionising ion level wave number (cm-1)
c                               1st dim: ionising ion level index
c          (i*4)  nmet_i      = number of ionising ion metastables
c          (i*4)  imeta_i()   = pointers to ionising metastables in full
c                               ionising ion state list
c                               1st dim: ionising metastable index
c          (i*4)  nte_ion()   = number of temperatures for direct ionis-
c                               ation data for initial metastable block
c                               1st dim: ionising ion metastable index
c          (r*8)  tea_ion(,)  = temperatures (K) for direct ionis-
c                               ation data for initial metastable block
c                               1st dim: ionising ion metastable index
c                               2nd dim: temperature index
c          (l*4)  lqred_ion(,)= .true. => direct ionisation data line
c                                         present for ionised ion state
c                               .false.=> data line not present for
c                                         ionised ion state.
c                               1st dim: ionising ion metastable index
c                               2nd dim: ionised ion state index
c          (r*8)  qred_ion(,,)= reduced direct ionisation rate coeffts.
c                               1st dim: ionising ion metastable index
c                               2nd dim: ionised ion state index
c                               3rd dim: temperature index
c          (i*4)  nf_a()      = number of Auger ionised ion final states
c                               1st dim: ionising ion metastable index
c          (i*4)  indf_a(,)   = Auger ionised ion final state
c                               1st dim: ionising ion metastable index
c                               2nd dim: final state index
c          (l*4)  lyld_a(,)   = .true. => Auger data for ionising ion excited state
c                               .false.=> no Auger data
c                               1st dim: ionising ion metastable index
c                               2nd dim: initial state index
c          (r*8)  yld_a(,,)   = Auger yields
c                               1st dim: ionising ion metastable index
c                               2nd dim: ionising ion excited state index
c                               3rd dim: ionised ion state index
```

```
c          (i*4)  nte_exc()    = number of temperatures for excitation
c                                data for initial metastable block
c                                1st dim: ionising ion metastable index
c          (r*8)  tea_exc(,)   = temperatures (K) for direct excit-
c                                ation data for initial metastable block
c                                1st dim: ionising ion metastable index
c                                2nd dim: temperature index
c          (l*4)  lqred_exc(,)= .true. => direct excitation data line
c                                          present for excited ion state
c                                .false.=> data line not present for
c                                          excited ion state.
c                                1st dim: ionising ion metastable index
c                                2nd dim: excited ionising ion state index
c          (r*8)  qred_exc(,,)= reduced excitation rate coeffts.
c                                1st dim: ionising ion metastable index
c                                2nd dim: excited ionising ion state index
c                                3rd dim: temperature index
c          (l*4)  l_ion()      = .true. => ionisation data present for metastable
c                                .false.=> ionisation data not present
c                                1st dim: ionising ion metastable index
c          (l*4)  l_aug()      = .true. => Auger data present for metastable
c                                .false.=> Auger data not present
c                                1st dim: ionising ion metastable index
c          (l*4)  l_exc()      = .true. => excitation data present for metastable
c                                .false.=> excitation data not present
c                                1st dim: ionising ion metastable index
c          (i*4)  ntext        = number of commment text lines
c          (c*80) ctext()      = comment text lines
c                                1st dim: index of text lines
c
c
c  routines:
c          routine    source    brief description
c          ------------------------------------------------------------
c          i4unit     adas      fetch unit number for output of messages
c          i4eiz0     adas      fetch nuclear charge for element symbol
c          xfesym     adas      fetch element symbol for nuclear charge
c          xxcase     adas      convert string to lower or upper case
c          xxhkey     adas      extract a key name value from a string
c          xxlast     adas      find last occurence of char in string
c          xxslen     adas      find first and last characters of string
c          xxdtes     adas      detect if config string is eissner/standard
c          xxcftr     adas      covert config string between eissner/standard
c
c  author:  Hugh Summers
c  date  :  22-05-2008
c
c
c  version  : 1.1
c  date     : 22-05-2008
c  modified : Hugh Summers
c               - first version
c
c
c-----------------------------------------------------------------------
      CHARACTER           CODE_F(NDLEV),            CODE_I(NDLEV)
      CHARACTER*(*)       CSTRGA_F(NDLEV),          CSTRGA_I(NDLEV)
      CHARACTER*80        CTEXT(NDTEXT)
      CHARACTER*2         CTYPE,       SEQ
      INTEGER             IA_F(NDLEV), IA_I(NDLEV), ILA_F(NDLEV)
      INTEGER             ILA_I(NDLEV),             IMETA_F(NDMET)
```

```
INTEGER             IMETA_I(NDMET),           INDF_A(NDMET,NDLEV)
INTEGER             ISA_F(NDLEV),             ISA_I(NDLEV)
INTEGER             IUNIT,       IZ,          IZ0,         IZ1
INTEGER             NDLEV,       NDMET,       NDTEM,       NDTEXT
INTEGER             NF_A(NDMET), NLVL_F,      NLVL_I,      NMET_F
INTEGER             NMET_I,      NTEXT,       NTE_EXC(NDMET)
INTEGER             NTE_ION(NDMET)
LOGICAL             LCSTRG_F,    LCSTRG_I,    LMET_F,      LMET_I
LOGICAL             LQRED_EXC(NDMET,NDLEV)
LOGICAL             LQRED_ION(NDMET,NDLEV),   LYLD_A(NDMET,NDLEV)
LOGICAL             L_AUG(NDMET),             L_EXC(NDMET)
LOGICAL             L_ION(NDMET)
REAL*8              BWNO_F,      BWNO_I
REAL*8              QRED_EXC(NDMET,NDLEV,NDTEM)
REAL*8              QRED_ION(NDMET,NDLEV,NDTEM)
REAL*8              TEA_EXC(NDMET,NDTEM),     TEA_ION(NDMET,NDTEM)
REAL*8              WA_F(NDLEV), WA_I(NDLEV), XJA_F(NDLEV)
REAL*8              XJA_I(NDLEV)
REAL*8              YLD_A(NDMET,NDLEV,NDLEV)
```

9.108 xxdata_24: Subroutine xxdata_24 from library adaslib

```
       subroutine xxdata_24( iunit  , dsname ,
     &                       nstore , nedim  ,
     &                       esym   , iz0    ,
     &                       nbsel  , isela  ,
     &                       iz     , iz1    ,
     &                       cdonor , crecvr , cfstat , ctype,
     &                       alph0  ,
     &                       iea    ,
     &                       teea   , scx
     &                     )
C-----------------------------------------------------------------------
C
C  ***************** FORTRAN77 SUBROUTINE: xxdata_24 ******************
C
C  PURPOSE:  To  fetch  data  from  input charge exchange cross-section
C            data for  given donor and receiver ions.
C
C  CALLING PROGRAM: ADAS509/SSCX
C
C  DATA:    Up to 'nstore' sets (data-blocks) of data may be  read from
C           the file - each block forming a complete set of  cross-
C           section values for given collision energies.
C           Each data-block  is  analysed independently of any  other
C           data-block.
C
C           The units used in the data file are taken as follows:
C
C           Collision energies  : eV/amu
C           Cross-section       : cm**2
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  IUNIT    = UNIT TO WHICH INPUT FILE IS ALLOCATED.
C  INPUT : (C*44) DSNAME   = FILENAME OF DATA SET BEING READ
C
C  INPUT : (I*4)  NSTORE   = MAXIMUM NUMBER  OF  INPUT DATA-BLOCKS  THAT
C                            CAN BE STORED.
C  INPUT : (I*4)  NEDIM    = MAX NUMBER OF COLLISION ENERGIES ALLOWED
C
C  OUTPUT: (C*2)  ESYM     = READ - RECEIVING ION - ELEMENT SYMBOL
C  OUTPUT: (I*4)  IZ0      = READ - RECEIVING ION - NUCLEAR CHARGE
C
C  OUTPUT: (I*4)  NBSEL    = NUMBER OF DATA-BLOCKS ACCEPTED & READ IN.
C  OUTPUT: (I*4)  ISELA()  = READ - DATA-SET DATA-BLOCK ENTRY INDICES
C                            DIMENSION: DATA-BLOCK INDEX
C
C  OUTPUT: (I*4)  IZ       = READ - RECEIVED ION -  CHARGE
C  OUTPUT: (I*4)  IZ1      = READ - RECEIVING ION - CHARGE
C
C  OUTPUT: (C*9)  CDONOR() = READ - DONOR ION IDENTIFICATION
C                            DIMENSION: DATA-BLOCK INDEX
C  OUTPUT: (C*9)  CFCODE() = READ - RECEIVER ION IDENTIFICATION
C                            DIMENSION: DATA-BLOCK INDEX
C  OUTPUT: (C*10) CFSTAT() = READ - FINAL STATE SPECIFICATION
C                            DIMENSION: DATA-BLOCK INDEX
C  OUTPUT: (C*2)  CTYPE()  = READ - CROSS-SECTION TYPE
C                            DIMENSION: DATA-BLOCK INDEX
C
C  OUTPUT: (R*8)  ALPH0()  = READ - LOW ENERGY EXTRAPOLATION PARM.
```

```
C                             DIMENSION: DATA-BLOCK INDEX
C
C
C  OUTPUT: (I*4)  IEA()     = READ - NUMBER OF COLLISION ENERGIES
C                             DIMENSION: DATA-BLOCK INDEX
C
C  OUTPUT: (R*8)  TEEA(,)   = READ - COLLISION ENERGIES (UNITS: eV/AMU)
C                             1st DIMENSION: COLLISION ENERGY INDEX
C                             2nd DIMENSION: DATA-BLOCK INDEX
C
C  OUTPUT: (R*8)  SCX(,)    =READ - FULL SET OF COLLISION CROSS-
C                                    SECTION VALUES (cm**2)
C                             1st DIMENSION: COLLISION ENERGY INDEX
C                             2nd DIMENSION: DATA-BLOCK INDEX
C
C          (I*4)  I4EIZ0    = FUNCTION - (SEE ROUTINES SECTION BELOW)
C          (I*4)  I4FCTN    = FUNCTION - (SEE ROUTINES SECTION BELOW)
C          (I*4)  I4UNIT    = FUNCTION - (SEE ROUTINE SECTION BELOW)
C          (I*4)  IBLK      = ARRAY INDEX: DATA-BLOCK INDEX
C          (I*4)  ITT       = ARRAY INDEX: COLLISION ENERGY INDEX
C          (I*4)  NENUM     = NUMBER OF COLLISION ENERGIES FOR CURRENT
C                             DATA-BLOCK
C          (I*4)  IABT      = RETURN CODE FROM 'I4FCTN'
C          (I*4)  IPOS1     = GENERAL USE STRING INDEX VARIABLE
C          (I*4)  IPOS2     = GENERAL USE STRING INDEX VARIABLE
C
C          (R*8)  R8FCTN    = FUNCTION - (SEE ROUTINES SECTION BELOW)
C
C          (L*4)  LBEND     = IDENTIFIES WHETHER THE LAST OF THE  INPUT
C                             DATA SUB-BLOCKS HAS BEEN LOCATED.
C                             (.TRUE. => END OF SUB-BLOCKS REACHED)
C
C          (C*1)  CSLASH    = '/' - DELIMITER FOR 'XXHKEY'
C          (C*2)  C2        = GENERAL USE TWO BYTE CHARACTER STRING
C          (C*3)  CKEY1     = 'FST'    - INPUT BLOCK HEADER KEY
C          (C*4)  CKEY2     = 'TYPE'   - INPUT BLOCK HEADER KEY
C          (C*5)  CKEY3     = 'ALPH0'  - INPUT BLOCK HEADER KEY
C          (C*4)  CKEY4     = 'ISEL'  - INPUT BLOCK HEADER KEY
C          (C*10) C10       = GENERAL USE TEN BYTE CHARACTER STRING
C          (C*80) C80       = GENERAL USE 80 BYTE  CHARACTER  STRING  FOR
C                             THE INPUT OF DATA-SET RECORDS.
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          XXHKEY     ADAS      OBTAIN KEY/RESPONSE STRINGS FROM TEXT
C          I4EIZ0     ADAS      RETURNS Z0 FOR GIVEN ELEMENT SYMBOL
C          I4FCTN     ADAS      CONVERT CHARACTER STRING TO INTEGER
C          I4UNIT     ADAS      FETCH UNIT NUMBER FOR OUTPUT OF MESSAGES
C          R8FCTN     ADAS      CONVERT CHARACTER STRING TO REAL*8
C
C
C-----------------------------------------------------------------------
C
C NOTES: Based on e9data.for (H P Summers, 06/06/96).
C
C
C VERSION  : 1.1
C DATE     : 27-03-2006
C MODIFIED : Martin O'Mullane
C              - First version
```

```
C
C-----------------------------------------------------------------------
      CHARACTER*9         CDONOR(NSTORE)
      CHARACTER*10        CFSTAT(NSTORE)
      CHARACTER*9         CRECVR(NSTORE)
      CHARACTER*2         CTYPE(NSTORE)
      CHARACTER*44        DSNAME
      CHARACTER*2         ESYM
      INTEGER             IEA(NSTORE), ISELA(NSTORE),            IUNIT
      INTEGER             IZ,          IZ0,         IZ1,         NBSEL
      INTEGER             NEDIM,       NSTORE
      REAL*8              ALPH0(NSTORE),            SCX(NEDIM,NSTORE)
      REAL*8              TEEA(NEDIM,NSTORE)
```

9.109 xxdata_25: Subroutine xxdata_25 from library adaslib

```
      subroutine xxdata_25( iunit  , a25fmt , dsname ,
     &                      ndtem  , ndden  , ndrep  , ndcor , nddiel ,
     &                      nddef  , ndimp  , ndein  , ndzef ,
     &                      iz0    , iz1    , outfmt ,
     &                      exfile , cxfile ,
     &                      ndens  , id_ref , densa  , denpa , denimpa,
     &                      deniona,
     &                      ntemp  , it_ref , tea    , tpa   , timpa  ,
     &                      tiona  ,
     &                      nzef   , iz_ref , zefa   ,
     &                      nbeam  , ib_ref , bmena  , denha , bmfra  ,
     &                      nimp   , im_ref , zimpa  , amimpa, frimpa ,
     &                      ts     , w      , w1     ,
     &                      cion   , cpy    , nip    , intd  , iprs   ,
     &                      ilow   , ionip  , nionip , ilprs , ivdisp ,
     &                      nmin   , nmax   , imax   , nrep  , wbrep  ,
     &                      jdef   , def    ,
     &                      jcor   , cor    , jmax   , epsil , fij    ,
     &                      wij
     &                    )
c-----------------------------------------------------------------------
c
c  ***************** fortran77 subroutine: xxdata_25 ******************
c
c  purpose:  To fetch  data  from  an adf25 driver dataset.
c
c  Notes:
c
c  Subroutine:
c
c   input : (i*4)  iunit    = unit number for input adf01 file.
c   input : (c*8)  a25fmt   = subdirectory type of adf25 to be read.
c   input : (c*80) dsname   = file name of adf25 format to be read.
c
c   input : (i*4)  ndtem    = maximum number of electron temperatures
c   input : (i*4)  ndden    = maximum number of electron densities
c   input : (i*4)  ndrep    = maximum number of representative n-shells
c   input : (i*4)  ndcor    = maximum number of DR bethe corrections
c   input : (i*4)  nddiel   = maximum number of DR core transitions
c   input : (i*4)  nddef    = maximum number of quantum defects
c   input : (i*4)  ndimp    = maximum number of plasma impurities
c   input : (i*4)  ndein    = maximum number of beam energies
c   input : (i*4)  ndzef    = maximum number of z effectives
c
c   input : (i*4)  iz0      = nuclear charge of bundle-n ion
c   input : (i*4)  iz1      = recombining ion charge of bundle-n ion
c   input : (c*5)  outfmt   = format of output ADAS data format for final
c                             results
c   input : (c*80) cxfile   = file name for charge exchange data input
c   input : (c*80) exfile   = file name for map of proj. matrix output
c
c   input : (i*4)  ndens    = number of electron densities
c   input : (i*4)  id_ref   = reference electron density pointer in vectors
c   input : (i*4)  densa()  = plasma electron density vector (cm-3)
c                             1st dim: index of electron density
c   input : (i*4)  denpa()  = plasma H+ density vector (cm-3)
c                             1st dim: index of electron density
c   input : (i*4)  denimpa()= plasma mean impurity ion density (cm-3)
c                             1st dim: index of electron density
```

```
c   input : (i*4)  deniona()= total ion density (plasma+impurity) (cm-3)
c                            1st dim: index of electron density
c
c   input : (i*4)  ntemp    = number of electron temperatures
c   input : (i*4)  id_ref   = reference electron temp. pointer in vectors
c   input : (i*4)  tea()    = plasma electron temp. vector (K)
c                            1st dim: index of electron temperature
c   input : (i*4)  tpa()    = plasma H+ temp. vector (K)
c                            1st dim: index of electron temperature
c   input : (i*4)  timpa()  = plasma mean impurity ion temp (K)
c                            1st dim: index of electron temperature
c   input : (i*4)  tiona()  = mean ion temp (plasma+impurity) (K)
c                            1st dim: index of electron temperature
c
c   input : (i*4)  nzef     = number of plasma zeff
c   input : (i*4)  iz_ref   = reference zeff pointer in vector
c   input : (i*4)  zefa()   = plasma zeff vector
c                            1st dim: index of zeff
c
c   input : (i*4)  nbeam    = number of beam energies
c   input : (i*4)  ib_ref   = reference beam energy pointer in vectors
c   input : (i*4)  bmena()  = beam energy vector (ev/amu)
c                            1st dim: index of beam energies
c   input : (i*4)  denha()  = beam H+ density vector (cm-3)
c                            1st dim: index of beam energies
c   input : (i*4)  bmfra()  = fractions of beam at each energy
c                            1st dim: index of beam energies
c
c   input : (i*4)  nimp     = number of plasma impurities (excl.h+)
c   input : (i*4)  im_ref   = reference impurity pointer in vectors
c   input : (r*8)  zimpa()  = impurity species charge
c                            1st dim: index of impurity
c   input : (r*8)  amimpa() = atomic mass number of impurity species
c                            1st dim: index of impurity
c   input : (r*8)  frimpa() = fraction of impurity (normalised to 1)
c                            1st dim: index of impurity
c
c   input : (r*8)  ts       = external radiation field temperature (K)
c   input : (r*8)  w        = general radiation dilution factor
c   input : (i*4)  w1       = external radiation field dilution factor
c                            for photo-ionisation form the ground level.
c
c   input : (r*8)  cion     = adjustment multiplier for ground ionis.
c   input : (r*8)  cpy      = adjustment multiplier for VR xsects.
c   input : (i*4)  nip      = range of delta n for IP xsects. (le.4)
c   input : (i*4)  intd     = order of Maxw. quad. for IP xsects.(le.3)
c   input : (i*4)  iprs     = 0  => default to VR xsects. beyond nip range
c                            1  => use PR xsects. beyond nip range
c   input : (i*4)  ilow     = 0  => no special low level data accessed
c                            1  => special low level data accessed
c   input : (i*4)  ionip    = 0 =>   no ion impact collisions included
c                            1 =>ion impact excit. and ionis. included
c   input : (i*4)  nionip   = range of delta n for ion impact
c                            excitation xsects.
c   input : (i*4)  ilprs    = 0 => default to vainshtein xsects.
c                            1 => use lodge-percival-richards xsects.
c   input : (i*4)  ivdisp   = 0 => ion impact at thermal Maxw. energies
c                            1 => ion impact at displaced thermal
c                                 energies according to the neutral
c                                 beam energy parameter
c                              * if(ivdisp=0 then special low level
```

```
c                                 data for ion impact is not substituted -
c                                 only vainshtein and lodge et al.
c                                 options are open.  Electron impact
c                                 data substitution does occur.
c   input : (i*4)  nmin     = lowest n-shell for population structure
c   input : (i*4)  nmax     = highest n-shell for population structure
c   input : (i*4)  imax     = number of representative n-shells
c   input : (i*4)  nrep()   = representative n-shells
c                             1st dim: index of representative n-shell
c   input : (r*8)  wbrep()  = dilution factors for nmin->nrep() trans.
c                             1st dim: index of representative n-shell
c   input : (i*4)  jdef     = number of n-shell quantum defects
c   input : (r*8)  def()    = quantum defects for n-shells
c                             1st dim: index of n-shell quantum defects
c                                      upwards from nmin
c   input : (i*4)  jcor     = number of DR Bethe correction factors
c   input : (r*8)  cor()    = DR Bethe correction factors
c                             1st dim: index of correction factor
c   input : (i*4)  jmax     = number of DR core transitions
c   input : (r*8)  epsil()  = reduced energy of core transition
c                             [delta Eij/I_H=(z+1)^2*epsil()]
c                             1st dim: index of DR core transition
c   input : (r*8)  fij()    = absorption oscillator strength for
c                             DR core transition
c                             1st dim: index of DR core transition
c   input : (r*8)  wij()    = dilution factor for DR core transition
c                             1st dim: index of DR core transition
c
c  Routines:
c
c     routine    source    brief description
c     ------------------------------------------------------------
c     xxcase     adas      convert string to upper or lower case
c     xxslen     adas      locate first and last char. of string
c
c
c  Author   : Hugh Summers
c  Date     : 23-05-2007
c
c
c  Version  : 1.1
c  Date     : 23-05-2007
c  Author   : Hugh Summers
c               - First release.
c
c  Version  : 1.2
c  Date     : 21-11-2007
c  Author   : Martin O'Mullane
c               - Increase number of lines to 150 to accommodate full
c                 range of energies/temperatures/densities of
c                 the dataformat.
c
c-----------------------------------------------------------------------
      CHARACTER*8         A25FMT
      CHARACTER*(*)       CXFILE,      DSNAME,      EXFILE
      CHARACTER*5         OUTFMT
      INTEGER             IB_REF,      ID_REF,      ILOW,        ILPRS
      INTEGER             IMAX,        IM_REF,      INTD,        IONIP
      INTEGER             IPRS,        IT_REF,      IUNIT,       IVDISP
      INTEGER             IZ0,         IZ1,         IZ_REF,      JCOR
      INTEGER             JDEF,        JMAX,        NBEAM,       NDCOR
```

```
INTEGER            NDDEF,       NDDEN,       NDDIEL,      NDEIN
INTEGER            NDENS,       NDIMP,       NDREP,       NDTEM
INTEGER            NDZEF,       NIMP,        NIONIP,      NIP
INTEGER            NMAX,        NMIN,        NREP(NDREP+1)
INTEGER            NTEMP,       NZEF
REAL*8             AMIMPA(NDIMP),            BMENA(NDEIN)
REAL*8             BMFRA(NDEIN),             CION
REAL*8             COR(NDCOR),  CPY,         DEF(NDDEF)
REAL*8             DENHA(NDEIN),             DENIMPA(NDDEN)
REAL*8             DENIONA(NDDEN),           DENPA(NDDEN)
REAL*8             DENSA(NDDEN),             EPSIL(NDDIEL)
REAL*8             FIJ(NDDIEL), FRIMPA(NDIMP)
REAL*8             TEA(NDTEM),  TIMPA(NDTEM)
REAL*8             TIONA(NDTEM),             TPA(NDTEM),  TS
REAL*8             W,           W1,          WBREP(NDREP)
REAL*8             WIJ(NDDIEL), ZEFA(NDZEF), ZIMPA(NDIMP)
```

9.110 xxdata_35: Subroutine xxdata_35 from library adaslib

```
       subroutine xxdata_35( iunit   ,
     &                       ndedge  , ndpts  ,
     &                       iedge   , ipts   ,
     &                       edge    , energy , trans
     &                     )

C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: xxdata_35 ******************
C
C  PURPOSE:  To fetch data from an adf35 data set.
C
C  SUBROUTINE:
C
C  input : (i*4)  iunit    = unit to which input file is allocated
C  input : (i*4)  ndedge   = maximum number of energy edges
C  input : (i*4)  ndpts    = maximum number of energy points
C
C  output: (i*4)  iedge    = number of energy edges
C  output: (i*4)  ipts     = number of energy points
C  output: (r*8)  edge()   = energies of the edges (eV).
C  output: (r*8)  energy() = tabulated energies (eV).
C  output: (r*8)  trans()  = tabulated transmission.
C
C ROUTINES:
C          routine    source    brief description
C          ------------------------------------------------------------
C          i4unit     ADAS      fetch unit number for message output
C
C
C
C VERSION  : 1.1
C DATE     : 19-07-2003
C MODIFIED : Martin O'Mullane
C              - First version
C
C-----------------------------------------------------------------------
      INTEGER             IEDGE,       IPTS,        IUNIT,       NDEDGE
      INTEGER             NDPTS
      REAL*8              EDGE(NDEDGE),             ENERGY(NDPTS)
      REAL*8              TRANS(NDPTS)
```

9.111 xxdata_37: Subroutine xxdata_37 from library adaslib

```
       subroutine xxdata_37( iunit  ,
     &                       nemax  , ntmax  ,
     &                       title  , icateg , nenerg , nblock ,
     &                       nform1 , param1 , nform2 , param2 ,
     &      ea     , fa     , teff   , mode   ,
     &      median , filnam , filout , calgeb ,
     &                       ealgeb
     &                      )

C-----------------------------------------------------------------------
C
C  ****************** fortran77 subroutine: xxdata_37 *****************
C
C  purpose:  To fetch data from an adf37 data set and detect its main
C            characteristics.
C
C  calling program: various
C
C  input : (i*4)  iunit    = unit to which input file is allocated
C  input : (i*4)  nemax    = max no of energy points that can be read in
C  input : (i*4)  ntmax    = max no of effective temps that can be read in
C
C  output: (c*80) title    = header for file
C  output: (i*4)  icateg   = category of file
C                              1 => superposition
C                              2 => numerical
C  output: (i*4)  nenerg   = type 1 => number of distribution families
C                            type 2 => number of energy points
C  output: (i*4)  nblock   = type 1 => number of members in output family
C                            type 2 => number of effective temperatures
C  output: (i*4)  nform1   = type of threshold behaviour
C                              1 => cutoff
C                              2 => energy^param1
C  output: (r*8)  param1   = parameter of threshold form
C  output: (i*4)  nform2   = type of high-energy behaviour
C                              1 => cutoff
C                              2 => energy^-param2(1)
C                              3 => exp(-param2(1)*energy)
C                              4 => exp(-param2(1)*energy^param2(2))
C  output: (r*8)  param2() = parameter of high-energy form
C  output: (r*8)  ea(,)    = energy points of tabulation
C  output: (r*8)  fa(,)    = distribution function tabulation
C  output: (r*8)  teff()   = effective temperature (eV)
C  output: (r*8)  mode()   = most probable energy (eV)
C  output: (r*8)  median() = median energy (eV)
C  output: (c*120)filnam() = file names of input families
C  output: (c*120)filout   = file name of output family
C  output: (c*25) calgeb(,)= distribution function algebra
C  output: (c*25) ealgeb() = energy parameter algebra
C
C  local : (i*4)  ieunit   = energy units of distribution function
C                              1 => kelvin
C                              2 => eV
C  local : (i*4)  i        = general use
C  local : (i*4)  j        = general use
C  local : (i*4)  med_index= energy index of median
C  local : (i*4)  mode_index()  = energy index of mode
C  local : (i*4)  dummy    = general use
C  local : (i*4)  ie       = general use
```

```
C  local : (i*4)  iblock   = general use
C  local : (r*8)  sum      = average energy contribution from i -> i+1
C  local : (r*8)  contrib()= average energy contribution from i -> i+1
C  local : (r*8)  de       = energy step from i -> i+1
C  local : (i*4)  ifirst   = position of first non-blank character in string
C  local : (i*4)  ilast    = position of last non-blank character in string
C  local : (i*4)  indx()   = index of algebra
C  local : (c*80) blank    = dummy string
C
C  routines:
C          routine    source    brief description
C          ------------------------------------------------------------
C          i4unit     ADAS      fetch unit number for output of messages
C          xxslen     ADAS      finds string length excluding leading and
C                               trailing blanks
C
C  author: Paul Bryans, University of Strathclyde
C
C  date:   04/02/04
C
C  update:
C
C
C-----------------------------------------------------------------------
      CHARACTER*25        CALGEB(NTMAX,4),          EALGEB(NTMAX)
      CHARACTER*120       FILNAM(NEMAX),            FILOUT
      CHARACTER*80        TITLE
      INTEGER             ICATEG,      IUNIT,       NBLOCK,      NEMAX
      INTEGER             NENERG,      NFORM1,      NFORM2,      NTMAX
      REAL*8              EA(NTMAX,NEMAX),          FA(NTMAX,NEMAX)
      REAL*8              MEDIAN(NTMAX),            MODE(NTMAX), PARAM1
      REAL*8              PARAM2(2),   TEFF(NTMAX)
```

9.112 xxdata_40: Subroutine xxdata_40 from library adaslib

```
       subroutine xxdata_40( iunit  , dsname  ,
     &                       nstore , ndpix   , ntdim  , nddim  ,
     &                       ndptnl , ndptn   , ndptnc , ndcnct ,
     &                       ndstack, ndcmt   ,
     &                       iz0    , is      , is1    , esym   ,
     &                       nptnl  , nptn    , nptnc  ,
     &                       iptnla , iptna   , iptnca ,
     &                       ncnct  , icnctv  ,
     &                       ncptn_stack      , cptn_stack      ,
     &                       lres   , lptn    , lcmt   , lsup   ,
     &                       nbsel  , isela   ,
     &                       npixa  , cfile   , ctype  , cindm  ,
     &                       ispbr  , isppr   , isstgr , ilzr   , ihzr ,
     &                       wvmina , wvmaxa  ,
     &                       ita    , ida     ,
     &                       teta   , teda    ,
     &                       fpec   , fpec_max,
     &                       ncmt_stack       , cmt_stack
     &                     )
c-----------------------------------------------------------------------
c
c  ***************** fortran77 subroutine: xxdata_40 ******************
c
c  purpose:  To fetch  data  from  an input feature photon emissivity
c            file for a given emitting element superstage .
c
c  calling programs: adas416/dxdata_40
c
c  data:    Up to 'nstore' sets (data-blocks) of data may be  read from
c           the file - each block forming a complete feature photon
c           emissivity coefft. for given temp/density grid and wave-.
c           length range.  Each data-block  is  analysed independently
c           of any  other datablock.
c
c           the units used in the data file are taken as follows:
c
c           temperatures : ev
c           densities    : cm-3
c           pec          : phot. cm3 s-1 pixel-1
c
c  subroutine:
c
c  input : (i*4)  iunit    = unit to which input file is allocated.
c          (i*4)  dsname   = name of opened data set on iunit
c
c          (i*4)  nstore   = maximum number  of  input data-blocks  that
c                            can be stored.
c          (i*4)  npix    = maximum number  of  pixels in a data-blocks
c                            that can be stored.
c          (i*4)  ntdim    = max number of electron temperatures allowed
c          (i*4)  nddim    = max number of electron densities    allowed
c          (i*4)  ndptnl   = maximum level of partitions
c          (i*4)  ndptn    = maximum no. of partitions in one level
c          (i*4)  ndptnc   = maximum no. of components in a partition
c          (i*4)  ndcnct   = maximum number of elements in connection
c          (i*4)  ndstack  = maximum number of partition text lines
c          (i*4)  ndcmt    = maximum number of comment text lines
c                             vector
c output:  (i*4)  iz0      = read - emitting ion - nuclear charge
```

```
c          (i*4)  is       = read - emitting ion - charge
c                            (generalised to superstage label)
c          (i*4)  is1      = read - emitting ion - charge + 1
c                            (generalised to superstage index= is + 1)
c          (c*2)  esym     = read - emitting ion - element symbol
c
c          (i*4)  nptnl    = number of partition levels in block
c          (i*4)  nptn()   = number of partitions in partition level
c                            1st dim: partition level
c          (i*4)  nptnc(,) = number of components in partition
c                            1st dim: partition level
c                            2nd dim: member partition in partition level
c          (i*4)  iptnla() = partition level label (0=resolved root,1=
c                                                   unresolved root)
c                            1st dim: partition level index
c          (i*4)  iptna(,) = partition member label (labelling starts at 0)
c                            1st dim: partition level index
c                            2nd dim: member partition index in partition
c                            level
c          (i*4)  iptnca(,,)= component label (labelling starts at 0)
c                            1st dim: partition level index
c                            2nd dim: member partition index in partition
c                            level
c                            3rd dim: component index of member partition
c          (i*4)  ncnct     = number of elements in connection vector
c          (i*4)  icnctv()  = connection vector of number of partitions
c                             of each superstage in resolved case
c                             including the bare nucleus
c                             1st dim: connection vector index
c          (i*4)  ncptn_stack = number of text lines in partition block
c          (c*80) cptn_stack()= text lines in partition block
c                               1st dim: text line index (1->ncptn_stack)
c
c          (l*4)  lres      = .true.  => partial file
c                           = .false. => not partial file
c          (l*4)  lptn      = .true.  => partition block present
c                           = .false. => partition block not present
c          (l*4)  lcmt      = .true.  => comment text block present
c                           = .false. => comment text block not present
c          (l*4)  lsup      = .true.  => ss use of filmem field
c                           = .false. => old use of filmem field
c
c          (i*4)  nbsel     = number of data-blocks accepted & read in.
c          (i*4)  isela()   = read - data-set data-block entry indices
c                            dimension: data-block index
c
c          (i*4)  npixa()   = number of pixels for data block
c                             1st dim: data-block index
c          (c*8)  cfile()   = specific ion file source string in older
c                             forms.  Field not present in superstage
c                             version, but reused for added information
c                             1st dim: data-block index
c          (c*8)  ctype()   = data type string
c                             1st dim: data-block index
c          (c*2)  cindm()   = metastable index string
c                             1st dim: data-block index
c
c          (i*4)  isppr()   = parent index for each feature block
c                              1st dim: index of block in adf40 file
c          (i*4)  ispbr()   = base index for each feature block
c                              1st dim: index of block in adf40 file
```

```
c          (i*4)  isstgr()  = s1 for each resolved data block
c                             1st dim: index of block in adf40 file
c          (i*4)  ilzr()    = lowest ion charge relating to feature
c                             1st dim: index of block in adf40 file
c          (i*4)  ihzr()    = highest ion charge relating to feature
c                             1st dim: index of block in adf40 file
c
c          (r*8)  wvmina()  = lowest wavelength of feature block
c                             dimension: data-block index
c          (r*8)  wvmaxa()  = highest wavelength of feature block
c                             dimension: data-block index
c
c          (i*4)  ita()     = number of electron temperatures
c                             dimension: data-block index
c          (i*4)  ida()     = read - number of electron densities
c                             1st dim: data-block index
c
c          (r*8)  teta(,)   =  electron temperatures (units: ev)
c                             1st dim: electron temperature index
c                             2nd dim: data-block index
c          (r*8)  teda(,)   = electron densities (units: cm-3)
c                             1st dim: electron density index
c                             2nd dim: data-block index
c
c          (r*8)  fpec(,,,) = feature photon emissivity coeffts
c                             1st dim: pixel index
c                             2nd dim: electron temperature index
c                             3rd dim: electron density index
c                             4th dim: data-block index
c          (r*8)  fpec_max()= feature photon emissivity coefft. power
c                             integral maximum (over wavelength interval)
c                             as a function of Te at first Ne value
c                             1st dim: data-block index
c          (i*4)  ncmt_stack = number of text lines in comment block
c          (c*80) cmt_stack()= text lines in comment block
c                               1st dim: text line index (1->ncmt_stack)
c
c routine: (i*4)  i4eiz0   = function - (see routines section below)
c          (i*4)  i4fctn   = function - (see routines section below)
c          (i*4)  i4unit   = function - (see routines section below)
c          (i*4)  iblk     = array index: data-block index
c          (i*4)  itt      = array index: electron temperature index
c          (i*4)  itd      = array index: electron density     index
c          (i*4)  ntnum    = number of electron temperatures for current
c                            data-block
c          (i*4)  ndnum    = number of electron densities    for current
c                            data-block
c          (i*4)  iabt     = return code from 'i4fctn'
c          (i*4)  ipos1    = general use string index variable
c          (i*4)  ipos2    = general use string index variable
c
c          (l*4)  lbend    = identifies whether the last of the  input
c                            data sub-blocks has been located.
c                            (.true. => end of sub-blocks reached)
c
c          (c*1)  cslash   = '/' - delimiter for 'xxhkey'
c          (c*2)  c2       = general use two byte character string
c          (c*5)  ionnam   = emitting ion read from dataset
c          (c*6)  ckey1    = 'filmem' - input block header key
c          (c*4)  ckey2    = 'type  ' - input block header key
c          (c*4)  ckey3    = 'indm  ' - input block header key
```

```
c          (c*4)  ckey4    = 'isel  ' - input block header key
c          (c*80) c80      = general use 80 byte  character  string  for
c                            the input of data-set records.
c
c routines:
c          routine    source    brief description
c          ------------------------------------------------------------
c          i4eiz0     adas      returns z0 for given element symbol
c          i4fctn     adas      convert character string to integer
c          i4unit     adas      fetch unit number for output of messages
c          r8fctn     adas      convert string to real number
c          xxmkrp     adas      make up root partition text lines
c          xxcase     adas      convert a string to upper or lower case
c          xxhkey     adas      obtain key/response strings from text
c          xxrptn     adas      analyse an adf11 file partition block
c          xxword     adas      extract position of number in buffer
c          xxslen     adas      find string less front and tail blanks
c
c author:  h. p. summers, university of strathclyde
c          ja7.08
c          tel. 0141-548-4196
c
c date:    13/06/06
c
c
c version  : 1.1
c date     : 25-11-2004
c modified : martin o'mullane
c              - first version
c
c version  : 1.2
c date     : 29-11-2004
c modified : martin o'mullane
c              - faulty 1001 format statement.
c
c version  : 1.3
c date     : 15-05-2006
c modified : Hugh Summers
c              - complete rewrite for  operation with superstages and
c                partitions, made similar to xxdata_15.for .
c
c version  : 1.4
c date     : 06-11-2006
c modified : Allan Whiteford
c              - correction of indexing npixa by ipx rather than iblk.
c
c version  : 1.5
c date     : 15-01-2007
c modified : Hugh Summers
c              - corrected metastable count for Ne+0.
c
c-----------------------------------------------------------------------
      CHARACTER*8         CFILE(NSTORE)
      CHARACTER*2         CINDM(NSTORE)
      CHARACTER*80        CMT_STACK(NDCMT),         CPTN_STACK(NDSTACK)
      CHARACTER*8         CTYPE(NSTORE)
      CHARACTER*80        DSNAME
      CHARACTER*2         ESYM
      INTEGER             ICNCTV(NDCNCT),           IDA(NSTORE)
      INTEGER             IHZR(NSTORE),             ILZR(NSTORE)
      INTEGER             IPTNA(NDPTNL,NDPTN)
```

```
INTEGER            IPTNCA(NDPTNL,NDPTN,NDPTNC)
INTEGER            IPTNLA(NDPTNL),           IS,          IS1
INTEGER            ISELA(NSTORE),            ISPBR(NSTORE)
INTEGER            ISPPR(NSTORE),            ISSTGR(NSTORE)
INTEGER            ITA(NSTORE), IUNIT,       IZ0,         NBSEL
INTEGER            NCMT_STACK,  NCNCT,       NCPTN_STACK, NDCMT
INTEGER            NDCNCT,      NDDIM,       NDPIX,       NDPTN
INTEGER            NDPTNC,      NDPTNL,      NDSTACK
INTEGER            NPIXA(NSTORE),            NPTN(NDPTNL)
INTEGER            NPTNC(NDPTNL,NDPTN),      NPTNL,       NSTORE
INTEGER            NTDIM
LOGICAL            LCMT,        LPTN,        LRES,        LSUP
REAL*8             FPEC(NDPIX,NTDIM,NDDIM,NSTORE)
REAL*8             FPEC_MAX(NSTORE),         TEDA(NDDIM,NSTORE)
REAL*8             TETA(NTDIM,NSTORE),       WVMAXA(NSTORE)
REAL*8             WVMINA(NSTORE)
```

9.113 xxdata_42: Subroutine xxdata_42 from library adaslib

```
       subroutine xxdata_42( fname  ,
     &                       ndtem  , ndden   , ndwvl  , ndpix   , ndmet ,
     &                       dsnin  , dsnexp  , dsnflt ,
     &                       dsnpec , dsnfpec , dsnplt , dsnpltf ,
     &                       celem  , iz0     , iz1    , bwno    ,
     &                       lnormr , nmetr   , imetrr ,
     &                       lioselr, lhselr  , lrselr , liselr  ,
     &                       lnselr , lpselr  , zeffr  ,
     &                       lmetrr , ltsclr  , ldsclr , lbrdir  ,
     &                       numter , numdensr, numwvlr,
     &                       tine   , tinp    , tinh   , dine    , dinp  ,
     &                       npix   , wvmin   , wvmax  , aminr
     &                     )

C-----------------------------------------------------------------------
C
C  ***************** FORTRAN77 subroutine: xxdata_42 ******************
C
C  Purpose:  To fetch data from an adas810 driver set (adf42).
C
C  Calling program: ADAS810
C
C
C  Subroutine:
C
C  input : (C*80) fname   = Name of adf42 driver file
C  input : (I*4)  ndtem   = maximum number of electron temperatures
C  input : (I*4)  ndden   = maximum number of electron tdensities
C  input : (I*4)  ndwvl   = maximum number of wavelength intervals
C  input : (I*4)  ndpix   = maximum number of pixels per wvln. interval
C  input : (I*4)  ndmet   = maximum number of metastables
C
C  output: (C*80) dsnin   = input adf04 file for population analysis
C  output: (C*80) dsnexp  = input adf18/a17_p208 expansion file
C  output: (C*80) dsnflt  = input adf35 filter file for power data
C  output: (C*80) dsnpec  = output adf15 file for traditional pec data
C  output: (C*80) dsnfpec = output adf40 file for feature pec data
C  output: (C*80) dsnplt  = output adf11 file for line power
C  output: (C*80) dsnpltf = output adf11 file for filtered line power
C
C  output: (C*2)  celem   = element symbol.
C  output: (I*4)  iz      =  recombined ion charge read
C  output: (I*4)  iz0     =         nuclear charge read
C  output: (I*4)  iz1     = recombining ion charge read
C                           (note: iz1 should equal iz+1)
C  output: (R*8)  bwno    = ionisation potential (cm-1) of lowest parent
C
C  output: (I*4)  nte     = number of electron temperatures
C  output: (R*8)  tea()   = electron temperatures (k)
C  output: (I*4)  ndens   = number of electron densities
C  output: (R*8)  densa() = electron densities (cm-3)
C
C  output: (I*4)  nwvl    = wvaelength intervals
C  output: (R*8)  npix()  = numbre of pixels assigned to wavelength interval
C  output: (R*8)  wvmin() = lower limit of wavelength interval (ang)
C  output: (R*8)  wvmax() = upper limit of wavelength interval (ang)
C
C  output: (L*4)  lmetr   = .TRUE.  => identify and resolve metastables
C                         = .FALSE. => do not resolve metastables
```

```
C  output: (L*4)  ltscl   = .TRUE.  => input temperatures are z-scaled
C                         = .FALSE. => input temperatures not z-scaled
C  output: (L*4)  ldscl   = .TRUE.  => input densities are z-scaled
C                         = .FALSE. => input densities not z-scaled
C  output: (L*4)  lbrdi   = .TRUE.  => impose ion temperature broadening
C                         = .FALSE. => input densities not z-scaled
C
C
C ROUTINES:
C          Routine    Source    Brief Description
C          ------------------------------------------------------------
C          i4unit     ADAS      Fetch unit number for output of messages
C      xxflnm     ADAS   Checks for & replaces the ADASCENT environment variable.
C
C AUTHOR:  Martin O'Mullane
C
C DATE:    28-02-2003
C
C NOTES: Copied from haddat.for
C
C
C VERSION  : 1.1
C DATE     : 28-05-2003
C MODIFIED : Martin O'Mullane
C              - First version
C
C-----------------------------------------------------------------------
      CHARACTER*2         CELEM
      CHARACTER*80        DSNEXP,      DSNFLT,      DSNFPEC,     DSNIN
      CHARACTER*80        DSNPEC,      DSNPLT,      DSNPLTF,     FNAME
      INTEGER             IMETRR(NDMET),            IZ0,         IZ1
      INTEGER             NDDEN,       NDMET,       NDPIX,       NDTEM
      INTEGER             NDWVL,       NMETR,       NPIX(NDWVL)
      INTEGER             NUMDENSR,    NUMTER,      NUMWVLR
      LOGICAL             LBRDIR,      LDSCLR,      LHSELR,      LIOSELR
      LOGICAL             LISELR,      LMETRR,      LNORMR,      LNSELR
      LOGICAL             LPSELR,      LRSELR,      LTSCLR
      REAL*8              AMINR,       BWNO,        DINE(NDDEN)
      REAL*8              DINP(NDDEN), TINE(NDTEM), TINH(NDTEM)
      REAL*8              TINP(NDTEM), WVMAX(NDWVL)
      REAL*8              WVMIN(NDWVL),             ZEFFR
```

9.114 xxdate: Subroutine xxdate from library adaslib

```
      SUBROUTINE XXDATE ( DATE )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXDATE *********************
C
C  PURPOSE: GATHERS CURRENT DATE AS A 8 BYTE STRING FROM IDL VIA PIPE
C
C  CALLING PROGRAM: GENERAL USE
C
C
C  SUBROUTINE:
C
C
C OUTPUT:  (C*8)  DATE     = CURRENT DATE (AS 'DD/MM/YY')
C
C          (I*4)   PIPEIN   = PARAMETER = UNIT NUMBER FOR OUTPUT TO PIPE
C          (I*4)   PIPEOU   = PARAMETER = UNIT NUMBER FOR INPUT TO PIPE
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C
C AUTHOR:  ANDREW BOWEN (TESSELLA SUPPORT SERVICES PLC)
C
C DATE:    24/05/93
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      CHARACTER*8         DATE
```

9.115 xxdcon: Subroutine xxdcon from library adaslib

```
      SUBROUTINE XXDCON( INTYP, OUTTYP, IZ1, IDVAL, DIN, DOUT )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXDCON *********************
C
C  PURPOSE: TO CONVERT AN ARRAY OF DENSITIES INTO SPECIFIED UNITS
C
C  CALLING PROGRAM: GENERAL USE
C
C  SUBROUTINE:
C  INPUT :    (I*4)  INTYP   = 1 => 'DIN(array)' UNITS: CM-3
C                              = 2 => 'DIN(array)' UNITS: REDUCED
C  INPUT :    (I*4)  OUTTYP  = 1 => 'DOUT(array)' UNITS: CM-3
C                              = 2 => 'DOUT(array)' UNITS: REDUCED
C  INPUT :    (I*4)  IZ1     = RECOMBINING ION CHARGE (= Z+1).
C  INPUT :    (I*4)  IDVAL   = NUMBER OF DENSITIES IN 'DIN(array)'
C  INPUT :    (R*8)  DIN()   = INPUT DENSITIES (STATED UNITS)
C  OUTPUT:    (R*8)  DOUT()  = OUTPUT DENSITIES (STATED UNITS)
C
C             (I*4)  I       = GENERAL USE
C
C             (R*8)  Z1P7    = 'IZ1'**7
C             (R*8)  DCONV() = DENSITY CONVERSION PARAMETERS
C
C ROUTINES:  NONE
C
C NOTE:
C            DENSITY CONVERSION PARAMETERS:
C
C            INTYP = 1 ; DCONV(1) =>              CM-3  -> OUTPUT UNITS
C            INTYP = 2 ; DCONV(2) =>           REDUCED  -> OUTPUT UNITS
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 4569
C
C DATE:    04/02/91
C
C UPDATE:  07/08/91 - PE BRIDEN: CHANGED 'DBLE(IZ1**7)' TO 'DBLE(IZ1)**7
C                                TO AVOID INTEGER OVERFLOW IF IZ1>21.
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IDVAL,       INTYP,       IZ1,         OUTTYP
      REAL*8              DIN(IDVAL),  DOUT(IDVAL)
```

9.116 xxdeci: Subroutine xxdeci from library adaslib

```
       subroutine xxdeci(x,iw,id,ifail)
c-----------------------------------------------------------------------
c
c ****************** fortran 77 subroutine: dgwpec ********************
c
c  purpose: To decide the number of decimals which can be printed
c           after the decimal point in a fixed format with a specified
c           field length.
c
c
c subroutine:
c
c  input : (r*8)  x         = decimal number for prining in fixed format
c  input : (i*4)  iw        = field length including sign and dec. pt.
c
c  output: (i*4)  id        = number of decimals to print after dec. pt.
c  output  (i*4)  ifail     = 0 => variable is printable as fixed point
c                               in the specified field length
c                             1 => variable cannot be printed as fixed
c                                  point in the specified field length
c
c routines:
c          routine    source    brief description
c          ------------------------------------------------------------
c         i4unit     adas      fetch unit number for output of messages
c
c
c author:  h. p. summers, university of strathclyde
c          ja7.08
c          tel. 0141-548-4196
c
c date:    26/04/06
c
c version: 1.1                          date: 26/04/2006
c modified: hugh p summers
c               - first edition.
c
c-----------------------------------------------------------------------
      INTEGER             ID,          IFAIL,       IW
      REAL*8              X
```

9.117 xxder1: Subroutine xxder1 from library adaslib

```
      SUBROUTINE XXDER1(FCN,M,N,X,FVEC,FJAC,LDFJAC,INFO,IPVT,WA,
     &                  LWA)
C-----------------------------------------------------------------------
C
C ROUTINE: XXDER1 (MINPACK ROUTINE LMDER1)
C
C PURPOSE:
C     MINIMIZE THE SUM OF THE SQUARES OF M NONLINEAR FUNCTIONS IN N
C     VARIABLES BY A MODIFICATION OF THE LEVENBERG-MARQUARDT
C     ALGORITHM.
C
C     THIS IS DONE BY USING THE MORE GENERAL LEAST-SQUARES SOLVER
C     LMDER. THE USER MUST PROVIDE A SUBROUTINE WHICH CALCULATES
C     THE FUNCTIONS AND THE JACOBIAN.
C
C INPUT:
C       FCN IS THE NAME OF THE USER-SUPPLIED SUBROUTINE WHICH
C         CALCULATES THE FUNCTIONS AND THE JACOBIAN. FCN MUST
C         BE DECLARED IN AN EXTERNAL STATEMENT IN THE USER
C         CALLING PROGRAM, AND SHOULD BE WRITTEN AS FOLLOWS.
C
C         SUBROUTINE FCN(M,N,X,FVEC,FJAC,LDFJAC,IFLAG)
C         INTEGER M,N,LDFJAC,IFLAG
C         DOUBLE PRECISION X(N),FVEC(M),FJAC(LDFJAC,N)
C         ----------
C         IF IFLAG = 1 CALCULATE THE FUNCTIONS AT X AND
C         RETURN THIS VECTOR IN FVEC. DO NOT ALTER FJAC.
C         IF IFLAG = 2 CALCULATE THE JACOBIAN AT X AND
C         RETURN THIS MATRIX IN FJAC. DO NOT ALTER FVEC.
C         ----------
C         RETURN
C         END
C
C         THE VALUE OF IFLAG SHOULD NOT BE CHANGED BY FCN UNLESS
C         THE USER WANTS TO TERMINATE EXECUTION OF LMDER1.
C         IN THIS CASE SET IFLAG TO A NEGATIVE INTEGER.
C
C       M IS A POSITIVE INTEGER INPUT VARIABLE SET TO THE NUMBER
C         OF FUNCTIONS.
C
C       N IS A POSITIVE INTEGER INPUT VARIABLE SET TO THE NUMBER
C         OF VARIABLES. N MUST NOT EXCEED M.
C
C       LDFJAC IS A POSITIVE INTEGER INPUT VARIABLE NOT LESS THAN M
C         WHICH SPECIFIES THE LEADING DIMENSION OF THE ARRAY FJAC.
C
CC       TOL IS A NONNEGATIVE INPUT VARIABLE. TERMINATION OCCURS
CC         WHEN THE ALGORITHM ESTIMATES EITHER THAT THE RELATIVE
CC         ERROR IN THE SUM OF SQUARES IS AT MOST TOL OR THAT
CC         THE RELATIVE ERROR BETWEEN X AND THE SOLUTION IS AT
CC         MOST TOL.
C
C       WA IS A DOUBLE PREC. WORK ARRAY OF LENGTH LWA.
C
C       LWA IS A POSITIVE INTEGER INPUT VARIABLE NOT LESS THAN 5*N+M.
C
C I/O:
C       X IS AN ARRAY OF LENGTH N. ON INPUT X MUST CONTAIN
C         AN INITIAL ESTIMATE OF THE SOLUTION VECTOR. ON OUTPUT X
```

```
C         CONTAINS THE FINAL ESTIMATE OF THE SOLUTION VECTOR.
C
C OUTPUT:
C       FVEC IS AN OUTPUT ARRAY OF LENGTH M WHICH CONTAINS
C         THE FUNCTIONS EVALUATED AT THE OUTPUT X.
C
C       FJAC IS AN OUTPUT M BY N ARRAY. THE UPPER N BY N SUBMATRIX
C         OF FJAC CONTAINS AN UPPER TRIANGULAR MATRIX R WITH
C         DIAGONAL ELEMENTS OF NONINCREASING MAGNITUDE SUCH THAT
C
C                T     T           T
C               P *(JAC *JAC)*P = R *R,
C
C         WHERE P IS A PERMUTATION MATRIX AND JAC IS THE FINAL
C         CALCULATED JACOBIAN. COLUMN J OF P IS COLUMN IPVT(J)
C         (SEE BELOW) OF THE IDENTITY MATRIX. THE LOWER TRAPEZOIDAL
C         PART OF FJAC CONTAINS INFORMATION GENERATED DURING
C         THE COMPUTATION OF R.
C
C       INFO IS AN INTEGER OUTPUT VARIABLE. IF THE USER HAS
C         TERMINATED EXECUTION, INFO IS SET TO THE (NEGATIVE)
C         VALUE OF IFLAG. SEE DESCRIPTION OF FCN. OTHERWISE,
C         INFO IS SET AS FOLLOWS.
C
C         INFO = 0  IMPROPER INPUT PARAMETERS.
C
C         INFO = 1  ALGORITHM ESTIMATES THAT THE RELATIVE ERROR
C                   IN THE SUM OF SQUARES IS AT MOST TOL.
C
C         INFO = 2  ALGORITHM ESTIMATES THAT THE RELATIVE ERROR
C                   BETWEEN X AND THE SOLUTION IS AT MOST TOL.
C
C         INFO = 3  CONDITIONS FOR INFO = 1 AND INFO = 2 BOTH HOLD.
C
C         INFO = 4  FVEC IS ORTHOGONAL TO THE COLUMNS OF THE
C                   JACOBIAN TO MACHINE PRECISION.
C
C         INFO = 5  NUMBER OF CALLS TO FCN WITH IFLAG = 1 HAS
C                   REACHED 100*(N+1).
C
C         INFO = 6  TOL IS TOO SMALL. NO FURTHER REDUCTION IN
C                   THE SUM OF SQUARES IS POSSIBLE.
C
C         INFO = 7  TOL IS TOO SMALL. NO FURTHER IMPROVEMENT IN
C                   THE APPROXIMATE SOLUTION X IS POSSIBLE.
C
C       IPVT IS AN INTEGER OUTPUT ARRAY OF LENGTH N. IPVT
C         DEFINES A PERMUTATION MATRIX P SUCH THAT JAC*P = Q*R,
C         WHERE JAC IS THE FINAL CALCULATED JACOBIAN, Q IS
C         ORTHOGONAL (NOT STORED), AND R IS UPPER TRIANGULAR
C         WITH DIAGONAL ELEMENTS OF NONINCREASING MAGNITUDE.
C         COLUMN J OF P IS COLUMN IPVT(J) OF THE IDENTITY MATRIX.
C
C CALLING PROGRAM: GENERAL USE
C
C ROUTINES:
C-----------------------------------------------------------------------
C     NAME     SOURCE   PURPOSE
C-----------------------------------------------------------------------
C     FCN      USER     SEE ABOVE
C     LMDER    MINPACK  DOES THE CALCULATION. FOLLOWS LATER IN THIS FILE
```

```
C-----------------------------------------------------------------------
C
C AUTHOR: ARGONNE NATIONAL LABORATORY. MINPACK PROJECT. MARCH 1980.
C     BURTON S. GARBOW, KENNETH E. HILLSTROM, JORGE J. MORE
C
C DATE:   05-06-96
C
C VERSION 1.1                           DATE: 05-06-96
C MODIFIED: WILLIAM OSBORN
C           - FIRST COPIED FOR ADAS USE. REMOVED TOL FROM PARAMETERS
C             AND HARDWIRED AS DSQRT(DPMPAR(1))
C
C VERSION 1.2                           DATE: 10-04-2007
C MODIFIED : Allan Whiteford
C               - Modified documentation as part of automated
C   subroutine documentation preparation.
C
C-----------------------------------------------------------------------

      DOUBLE PRECISION    FJAC(LDFJAC,N),           FVEC(M),      WA(LWA)
      DOUBLE PRECISION    X(N)
      INTEGER             INFO,         IPVT(N),    LDFJAC,       LWA
      INTEGER             M,            N
      DOUBLE PRECISION    X(N)
      INTEGER             N
      DOUBLE PRECISION    DIAG(N),      FACTOR,     FJAC(LDFJAC,N)
      DOUBLE PRECISION    FTOL,         FVEC(M),    GTOL,         QTF(N)
      DOUBLE PRECISION    WA1(N),       WA2(N),     WA3(N),       WA4(M)
      DOUBLE PRECISION    X(N),         XTOL
      INTEGER             INFO,         IPVT(N),    LDFJAC,       M
      INTEGER             MAXFEV,       MODE,       N,            NFEV
      INTEGER             NJEV,         NPRINT
      DOUBLE PRECISION    DELTA,        DIAG(N),    PAR,          QTB(N)
      DOUBLE PRECISION    R(LDR,N),     SDIAG(N),   WA1(N),       WA2(N)
      DOUBLE PRECISION    X(N)
      INTEGER             IPVT(N),      LDR,        N
      DOUBLE PRECISION    A(LDA,N),     ACNORM(N),  RDIAG(N),     WA(N)
      INTEGER             IPVT(LIPVT),  LDA,        LIPVT,        M
      INTEGER             N
      LOGICAL             PIVOT
      DOUBLE PRECISION    DIAG(N),      QTB(N),     R(LDR,N)
      DOUBLE PRECISION    SDIAG(N),     WA(N),      X(N)
      INTEGER             IPVT(N),      LDR,        N
```

9.118 xxdsn2: Subroutine xxdsn2 from library adaslib

```
      SUBROUTINE XXDSN2 ( DSNIN   ,
     &                    DSNOUT  , LEXIST
     &                  )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXDSN2 *********************
C
C  PURPOSE: TO CHECK IF A FULLY QUALIFIED MVS SEQUENTIAL OR PARTITIONED
C           DATA SET EXISTS AND RETURNS THE DATA SET NAME IN A FORM FOR
C           DYNAMIC ALLOCATION.
C
C                   ***** 'DSNIN' MUST BE OF VALID FORM *****
C
C           NOTE: INPUT DSN MUST START WITH THE USERID AND MAY OR MAY
C                 NOT BE ENCLOSED IN QUOTES.  IT CAN ALREADY BE  IN A
C                 FORM SUITABLE FOR DYNAMIC  ALLOCATION IN WHICH CASE
C                 THE OUTPUT DSN AND INPUT DSN  ARE  IDENTICAL   (I.E.
C                 NEITHER HAVE QUOTES AND BOTH START WITH A '/'.)
C
C  CALLING PROGRAM: GENERAL USE
C
C  SUBROUTINE:
C
C  INPUT : (C*(*)) DSNIN   = DATASET NAME
C                            (CAN BE ENCLOSED WITHIN APOSTROPHES OR
C                             START WITH A '/')
C
C  OUTPUT: (C*(*)) DSNOUT  = VERSION OF  'DSNIN'  ALETRED SO THAT IT
C                            CAN BE USED TO DYNAMICALLY OPEN THE FILE.
C                            (NOTE: APOSTROPHES REMOVED, / ADDED AT
C                             START IF REQUIRED.)
C  OUTPUT: (L*4)   LEXIST  = .TRUE.  => FILE 'DSNIN' ALREADY EXISTS
C                          = .FALSE. => FILE 'DSNIN' DOES NOT EXISTS
C
C          (C*1)   QUOTE   = PARAMETER: APOSTROPHE (').
C          (C*1)   BSLASH  = PARAMETER: BACK SLASH (/).
C
C          (I*4)   ILEND   = LENGTH IN BYTES OF 'DSNIN'
C          (I*4)   I1      = LOCATION OF FIRST APOSTROPHE IN 'DSNIN'
C          (I*4)   I2      = LOCATION OF SECOND APOSTROPHE IN 'DSNIN'
C          (I*4)   IDSLEN  = IDENTIFIES THE NUMBER OF CHARACTERS PRESENT
C                            IN THE INPUT DATA SET NAME MINUS ONE.
C                            (I.E. NUMBER   OF   CHARACTERS  BETWEEN
C                                  APOSTROPHES IF PRESENT.)
C
C          (L*4)   LQUOTE  = .TRUE.  => APOSTROPHES PRESENT
C                            .FALSE. => NO APOSTROPHES PRESENT
C
C NOTE:
C
C          IDSLEN >=0 => NO APOSTROPHES IN 'DSNIN' OR TWO SEPERATED
C                        BY AT LEAST ONE CHARACTER.
C          IDSLEN =-1 => TWO ADJACENT APOSTROPHES
C          IDSLEN =-2 => ONLY ONE APOSTROPHE FOUND
C
C          IF:
C
C          IDSLEN =-1 ERROR: PROGRAM STOPS
C          IDSLEN =-2 ERROR: PROGRAM STOPS.
```

```
C
C
C ROUTINES:   NONE
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 4569
C
C DATE:    12/12/90  (IN ADAS90 VERSION 1 THIS ROUTINE IS CALLED XXDSN1)
C
C-----------------------------------------------------------------------
      CHARACTER*(*)       DSNIN,       DSNOUT
      LOGICAL             LEXIST
```

9.119 xxdtes: Subroutine xxdtes from library adaslib

```
      subroutine xxdtes( cstrg , leiss  , lstan  ,  nvlce )
c
c-----------------------------------------------------------------------
c
c  ****************** fortran77 subroutine: xxdtes *********************
c
c  purpose: Detects if the configuration string from a specific ion
c           level list line is of eissner form , standard form or
c           neither.
C
C           If neither, the subroutine checks to see if it is a
c           bundle (* in the string) or based on a parent ([..] in
c           the string).  If the string is of Eissner or standard
c           form, the n-shell and l-shell of the outermost
c           (valence)  electron is returned.
c
c           A version of this routine with a more extended return of
c           parameters and bale to handle very long configuration
c           strings is available as 'g5dtes.for'.
c
c  calling programs: general use
c
c  subroutine:
c
c  input : (c*(*)) cstrg    = configuration character string
c  output: (l*4)   leiss    = .true.  => eissner form
c                             .false. => not eissner form
c  output: (l*4)   lstan    = .true.  => standard form
c                             .false. => not standard form
c  output: (i*4)   nvlce    = outer electron n-shell if recognisable
c
c          (l*4)   lbndl    = .true.  => bundled form ('*' found)
c                             .false. => not bundled form
c          (l*4)   lprnt    = .true.  => parent form ('[...]' found)
c                             .false. => not parent form
c          (c*19)  cstr_top = leading part of config. string in Eissner
c                             format (no leading blank, trailing blanks)
c          (c*(*)) cstr_tail= trailing part of config. string in Eissner
c                             format (no leading blank, trailing blanks)
c          (i*4)   lvlce    = outer electron l-shell if recognisable
c
c          (i*4)   i       = general use
c          (i*4)   iabt    = return code (see specific function)
c                            0 => ok
c                            1 => fault detected
c          (i*4)   icfsel  = 1 => standard form out, standard form in
c                            2 => eissner  form out, standard form in
c                            3 => standard form out, eissner  form in
c                            4 => eissner  form out, eissner  form in
c          (i*4)   ishel   = shell counter
c          (i*4)   ip      = parity of configuration
c          (i*4)   maxn    = n_shell sum for configuration
c          (i*4)   nshel   = number of shells identified ffrom string
c          (i*4)   ndword  = maximum number of words in string
c          (i*4)   nfirst  = first word to be extracted from string
c          (i*4)   nwords  = number of words in string
c          (i*4)   nela()  = number of electrons in each shell
c          (i*4)   ifirst()= position of first char. of word in string
c          (i*4)   ilast() = position of last char. of word in string
```

```
c
c          (c*1)   cdelim  = separators for words in string
c          (c*19)  cstrgo  = general use string
c          (c*19)  strg    = standard form configuration string
c          (c*19)  strge   = eissner form configuration string
c          (c*1)   cheisa()= eissner character for orbitals
c          (c*2)   chstda()= standard orbital spec. for each shell
c          (c*2)   cnela() = chars. for no. of equiv. elec. in shell
c                            (eissner form case)
c          (c*1)   chqa()  = index to hexadecimal conversions
c          (c*1)   chra()  = char. for no. of. equiv. elec. in shell
c                            (standard form case)
c
c routines:
c          routine    source   brief description
c          ------------------------------------------------------------
c          i4fctn     adas     converts character string to integer
c          i4ngrp     adas     returns n quantum number in the
c                              eissner single hexadecimal character form
c          i4pgrp     adas     returns parity of orbital given the
c                              eissner single hexadecimal character form
c          i4schr     adas     returns numerical value for number of
c                              equivalent electrons given as hex> char.
c          cstgrp     adas     returns term of orbital given in the
c                              eissner single hexadecimal character form
c          ceigrp     adas     returns eissner code for orbital
c          xxword     adas     finds number of words in a string
c          xxcmps     adas     compare standard config. strings
c
c
c
c author:  h. p. summers, university of strathclyde
c          ja8.08
c          tel. 0141-553-4196
c
c date:    19/02/03
c
C VERSION: 1.1                          DATE: 19-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - PUT UNDER S.C.C.S. CONTROL
C
C VERSION: 1.2                          DATE: 14-10-96
C MODIFIED: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C               - ADDED CHANGES DATED 01/10/96 ABOVE
C
C VERSION: 1.3                          DATE: 28-08-97
C MODIFIED: HUGH SUMMERS
C               - ADDED CHANGES TO CHECK 'G' STATES
C
C VERSION: 1.4                          DATE: 19/02/03
C MODIFIED: HUGH SUMMERS
C               - Rewrite based on g5dtes.for
C
C VERSION: 1.5                          DATE: 28/09/2004
C MODIFIED: Martin O'Mullane
C               - Incorrect redirection when checking the Eissner pattern.
C                 The if statement block checking ir jumped out of the
C                 current sub-block to the end of the previous sub-block
C                 rather than to the end of its own sub-block.
C
C VERSION: 1.6                          DATE: 17/05/2007
```

```
C MODIFIED: Allan Whiteford
C           - Updated comments as part of subroutine documentation
C             procedure.
C
c-----------------------------------------------------------------------
c-----------------------------------------------------------------------
      CHARACTER*(*)       CSTRG
      INTEGER             NVLCE
      LOGICAL             LEISS,       LSTAN
```

9.120 xxecon: Subroutine xxecon from library adaslib

```
      SUBROUTINE XXECON( INTYP, OUTTYP, IEVAL, EIN, EOUT )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXECON *********************
C
C  PURPOSE: TO CONVERT AN ARRAY OF VELOCITIES/ENERGIES INTO A  SPECIFIED
C           FORM.
C
C  CALLING PROGRAM: GENERAL USE
C
C  SUBROUTINE:
C
C  INPUT :    (I*4)  INTYP   = 1 => 'EIN(array)' UNITS: AT. UNITS (VEL.)
C                              = 2 => 'EIN(array)' UNITS: CM/SEC    (VEL.)
C                              = 3 => 'EIN(array)' UNITS: EV/AMU  (ENERGY)
C  INPUT :    (I*4)  OUTTYP  = 1 => 'EOUT(array)' UNITS: AT.UNITS (VEL.)
C                              = 2 => 'EOUT(array)' UNITS: CM/SEC   (VEL.)
C                              = 3 => 'EOUT(array)' UNITS: EV/AMU (ENERGY)
C  INPUT :    (I*4)  IEVAL   = NO. OF VELOCITIES/ENERGIES IN EIN(array)
C  INPUT :    (R*8)  EIN()   = INPUT  VELOCITIES/ENERGIES (STATED UNITS)
C  OUTPUT:    (R*8)  EOUT()  = OUTPUT VELOCITIES/ENERGIES (STATED UNITS)
C
C             (R*8)  AMU2KG  = PARAMETER: AMU TO KG CONVERSION FACTOR
C             (R*8)  EV2J    = PARAMETER: EV TO JOULES CONVERSION FACTOR
C             (R*8)  M2CM    = PARAMETER: METRES TO CM CONVERSION FACTOR
C             (R*8)  VELE0H  = PARAMETER: ORBITAL VELOCITY  (CM/SEC)  OF
C                              AN ELECTRON IN THE SMALLEST  ORBIT  OF  A
C                              HYDROGEN ATOM (BOHR) = 2.1877D+8 CM/SEC
C             (R*8)  AT2VEL  = AT.UNITS (VEL) TO CM/SEC (VEL) CONVERSION
C             (R*8)  VEL2AT  = CM/SEC (VEL) TO AT.UNITS (VEL) CONVERSION
C             (R*8)  VEL2EN  = CM/SEC (VEL) TO EV/AMU (ENGY.) CONVERSION
C             (R*8)  EN2VEL  = EV/AMU (ENGY.) TO CM/SEC (VEL) CONVERSION
C             (R*8)  AT2EN   = AT.UNITS (VEL) TO EV/AMU (ENG) CONVERSION
C             (R*8)  EN2AT   = EV/AMU (ENG) TO AT.UNITS (VEL) CONVERSION
C
C             (I*4)  I       = GENERAL USE
C
C             (R*8)  ECONV() = ENERGY/VELOCITY CONVERSION PARAMETERS
C
C ROUTINES:  NONE
C
C NOTE:
C            ENERGY/VELOCITY CONVERSION PARAMETERS:
C
C            INTYP = 1 ; ECONV(1) =>  VELOCITY: AT.UNITS -> OUTPUT FORM
C            INTYP = 2 ; ECONV(2) =>  VELOCITY: CM/SEC   -> OUTPUT FORM
C            INTYP = 3 ; ECONV(3) =>  ENERGY  : EV/AMU   -> OUTPUT FORM
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 4569
C
C DATE:    05/02/91
C
C VERSION: 1.2 DATE: 01-05-96
C MODIFIED: WILLIAM OSBORN
C     - CORRECTED AT2EN. IT WAS AT2VEL*VEL2EN.
C
```

```
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IEVAL,       INTYP,       OUTTYP
      REAL*8              EIN(IEVAL),  EOUT(IEVAL)
```

9.121 xxeiam: Subroutine xxeiam from library adaslib

```
       subroutine xxeiam( esym  , xsym )
c-----------------------------------------------------------------------
c
c  ****************** fortran77 subroutine: xxeiam *********************
c
c  purpose:  Routine to return the atomic mass number for a given element
c            symbol esym.
c
c  calling program: gapecf
c
c
c  subroutine:
c
c  input : (i*4)  esym   = the symbol of the element required (note that
c                          case is unimportant as the uppercase of the
c                          input value is always taken).
c  output: (i*4)  xsym   = the float value of the atomic mass of the element.
c                          If the symbol is not recognised then the value
c                          of xsym returned is zero.
c
c routines:
c          routine    source    brief description
c          ------------------------------------------------------------
c          xxcase     adas      converts a string to upper or lower case
c
c author:  Hugh Summers, University of Strathclyde
c          JA7.08
c          tel. 0141-548-4196
c
c date:    15/01/02
c
c update:
c
c-----------------------------------------------------------------------
      CHARACTER*(*)       ESYM
      REAL*8              XSYM
```

9.122 xxeign: Subroutine xxeign from library adaslib

```
      SUBROUTINE XXEIGN(A,IA,N,RR,RI,VR,VI,IV1,FV1,IERR)
C-----------------------------------------------------------------------
C
C ROUTINE: XXEIGN
C
C PURPOSE: FINDS THE EIGENVALUES AND EIGENVECTORS OF A GENERAL REAL
C          MATRIX.
C
C          REPLACES NAG ROUTINE F02AGF ALTHOUGH THERE ARE SEVERAL
C          DIFFERENCES - YOU SHOULD COMPARE THE DOCUMENTATION. THIS
C          IS A FRONT-END TO THE NETLIB LIBRARY PROGRAM RG.F WHICH
C          FOLLOWS TOGETHER WITH ALL ITS DEPENDENCIES.
C
C      *** N.B. THE EIGENVECTORS ARE NOT NORMALIZED TO UNIT LENGTH ***
C
C CALLING PROGRAM: GENERAL USE
C
C INPUT:
C
C     (R*8)   A(,)    CONTAINS THE REAL GENERAL MATRIX.
C
C     (I*4)   IA      THE ROW DIMENSION OF THE TWO-DIMENSIONAL
C                     ARRAY PARAMETERS AS DECLARED IN THE CALLING
C                     PROGRAM DIMENSION STATEMENT.
C
C     (I*4)   N       THE ORDER OF THE MATRIX  A.
C     (I*4)   IV1()   WORK ARRAY, DIMENSION = N.
C     (R*4)   FV1()   WORK ARRAY, DIMENSION = N.
C
C OUTPUT:
C
C     (R*8)   RR()    REAL PART OF THE EIGENVALUES
C     (R*8)   RI()    IMAGINARY PART OF EIGENVALUES
C                     COMPLEX CONJUGATE PAIRS OF EIGENVALUES APPEAR
C                     CONSECUTIVELY WITH THE EIGENVALUE HAVING THE
C                     POSITIVE IMAGINARY PART FIRST.
C
C     (R*8)   VR(,)   REAL PARTS OF THE EIGENVECTORS. FIRST DIMENSION
C                     = IA.
C                     THE REAL PART OF THE EIGENVECTOR CORRESPONDING TO
C                     THE I-TH EIGENVALUE IS IN VR(J,I), J=1...N
C     (R*8)   VI(,)   IMAGINARY PARTS OF THE EIGENVECTORS. FIRST
C                     DIMENSION = IA
C                     THE IMAGINARY PART OF THE EIGENVECTOR
C                     CORRESPONDING TO THE I-TH EIGENVALUE IS IN
C                     VR(J,I), J=1...N
C     (I*4)   IERR    AN ERROR CODE SET TO ZERO IF NO ERROR. SEE HRQ AND
C                     HRQ2 FOR A DESCRIPTION OF THIS VARIABLE
C
C ROUTINES:
C-----------------------------------------------------------------------
C     NAME     SOURCE   PURPOSE
C-----------------------------------------------------------------------
C     RG       NETLIB   CALCULATES THE EIGENVALUES AND EIGENVECTORS
C     DPMPAR   ADAS/NETLIB   RETURNS MACHINE DEPENDANT PARAMETERS
C-----------------------------------------------------------------------
C
C AUTHOR: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC.)
C
```

```
C DATE:   10/06/96
C
C VERSION 1.1                              DATE: 10/06/96
C MODIFIED: WILLIAM OSBORN
C           - FIRST VERSION
C VERSION: 1.2      DATE:27/06/96
C MODIFIED: WILLIAM OSBORN
C     - REMOVED UNUSED VARIABLES
C
C VERSION: 1.3      DATE:27/06/96
C MODIFIED: WILLIAM OSBORN
C     - ADDED PDONE VARIABLE TO ELIMINATE THE NEED FOR ASSIGNMENT
C             TO A DO-LOOP COUNTER.
C
C VERSION: 1.4      DATE:17/05/07
C MODIFIED: Allan Whiteford
C           - Updated comments as part of subroutine documentation
C             procedure.
C
C-----------------------------------------------------------------------
C
      INTEGER             IA,           IERR,         IV1(N),       N
      REAL*8              A(IA,N),      FV1(N),       RI(N),        RR(N)
      REAL*8              VI(IA,N),     VR(IA,N)
      DOUBLE PRECISION    A(NM,N),      FV1(N),       WI(N),        WR(N)
      DOUBLE PRECISION    Z(NM,N)
      INTEGER             IERR,         IV1(N),       MATZ,         N
      INTEGER             NM
      DOUBLE PRECISION    A(NM,N),      SCALE(N)
      INTEGER             IGH,          LOW,          N,            NM
      DOUBLE PRECISION    SCALE(N),     Z(NM,M)
      INTEGER             IGH,          LOW,          M,            N
      INTEGER             NM
      DOUBLE PRECISION    AI,           AR,           BI,           BR
      DOUBLE PRECISION    CI,           CR
      DOUBLE PRECISION    A(NM,N)
      INTEGER             IGH,          INT(IGH),     LOW,          N
      INTEGER             NM
      DOUBLE PRECISION    A(NM,IGH),    Z(NM,N)
      INTEGER             IGH,          INT(IGH),     LOW,          N
      INTEGER             NM
      DOUBLE PRECISION    H(NM,N),      WI(N),        WR(N)
      INTEGER             IERR,         IGH,          LOW,          N
      INTEGER             NM
      DOUBLE PRECISION    H(NM,N),      WI(N),        WR(N),        Z(NM,N)
      INTEGER             IERR,         IGH,          LOW,          N
      INTEGER             NM
```

9.123 xxeiz0: Subroutine xxeiz0 from library adaslib

```
      SUBROUTINE XXEIZ0 ( ESYM , IZ0 )
C-----------------------------------------------------------------------
C
C ******************* FORTRAN77 SUBROUTINE: XXEIZ0 *********************
C
C PURPOSE: TO RETURN THE NUCLEAR CHARGE IZ0 FOR THE ELEMENT SYMBOL ESYM
C
C CALLING PROGRAM: GENERAL USE
C
C SUBROUTINE:
C
C  INPUT : (C*2)  ESYM    = SYMBOL OF ELEMENT WITH NUCLEAR CHARGE 'IZ0'
C  OUTPUT: (I*4)  IZ0     = ELEMENT NUCLEAR CHARGE
C
C          (I*4)  NSYM    = PARAMETER = NUMBER OF SYMBOLS LISTED
C
C          (I*4)  I       = GENERAL ARRAY USE
C
C          (C*2)  SYMBOL()= SYMBOLS OF FIRST 'NSYM' ELEMENTS.
C                           ARRAY DIMENSION => NUCLEAR CHARGE
C
C NOTES:    IF SYMBOL IS NOT RECOGNISED, I.E.NOT IN Z0 RANGE 1 & 'NSYM',
C           THEN THE INTEGER 'IZ0' IS RETURNED AS ZERO.
C
C ROUTINES: NONE
C
C
C AUTHOR:   PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C           K1/0/81
C           JET EXT. 4569
C
C DATE:     21/08/90
C
C UPDATE:     15/01/91 - PE BRIDEN - ADAS91 - ARGUMENTS REVERSED
C
C MODIFIED:
C
C VERSION:  1.1 TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)  14/09/95
C     -FIRST RELEASE (UNIX PORT)
C
C VERSION:  1.2 HUGH SUMMERS      17/09/99
C           -   INCREASED ELEMENT NUMBER TO 92
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      CHARACTER*2         ESYM
      INTEGER             IZ0
```

9.124 xxelem: Subroutine xxelem from library adaslib

```
      SUBROUTINE XXELEM ( IZ0 , ENAME )
C
C-----------------------------------------------------------------------
C
C ******************* FORTRAN77 SUBROUTINE: XXELEM *********************
C
C PURPOSE: TO RETURN THE NAME OF THE ELEMENT WITH NUCLEAR CHARGE IZ0
C
C CALLING PROGRAM: GENERAL USE
C
C SUBROUTINE:
C
C  INPUT : (I*4)  IZ0     = ELEMENT NUCLEAR CHARGE
C  OUTPUT: (C*12) ENAME   = NAME OF ELEMENT WITH NUCLEAR CHARGE 'IZ0'
C
C          (C*12) NAMES() = NAMES OF FIRST 92 ELEMENTS.
C                           ARRAY DIMENSION => NUCLEAR CHARGE
C
C NOTES:    IF NUCLEAR CHARGE IS OUT OF RANGE, I.E.NOT BETWEEN 1 & 92,
C           THEN THE CHARACTER STRING 'ENAME' IS RETURNED BLANK.
C
C ROUTINES: NONE
C
C
C AUTHOR:   PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C           K1/0/81
C           JET EXT. 4569
C
C DATE:     05/07/90
C
C UPDATE:   15/01/91 - PE BRIDEN - ADAS91 - ARGUMENTS REVERSED
C
C UPDATE:   25/01/91 - PE BRIDEN - ADAS91 - 'ENAME' & 'NAMES()' DECLARED
C                                           AS C*12 INSTEAD OF C*15.
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 21-03-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - PUT UNDER S.C.C.S. CONTROL
C
C VERSION: 1.2              DATE: 17-09-99
C            HUGH SUMMERS, UNIVERSITY OF STRATHCLYDE
C               - INCREASED ELEMENT NUMBER TO 92.
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      CHARACTER*12        ENAME
      INTEGER             IZ0
```

9.125 xxelz0: Subroutine xxelz0 from library adaslib

```
      SUBROUTINE XXELZ0 ( ELEM , IZ0 )
C-----------------------------------------------------------------------
C
C ******************* FORTRAN77 SUBROUTINE: XXEIZ0 *********************
C
C PURPOSE: TO RETURN THE NUCLEAR CHARGE IZ0 FOR THE ELEMENT NANE
C
C CALLING PROGRAM: GENERAL USE
C
C SUBROUTINE:
C
C  INPUT : (C*2)  ELEN    = ELEMENT NAME WITH NUCLEAR CHARGE 'IZ0'
C  OUTPUT: (I*4)  IZ0     = ELEMENT NUCLEAR CHARGE
C
C          (I*4)  NSYM    = PARAMETER = NUMBER OF SYMBOLS LISTED
C
C          (I*4)  I       = GENERAL ARRAY USE
C
C          (C*2)  SYMBOL()= SYMBOLS OF FIRST 'NSYM' ELEMENTS.
C                           ARRAY DIMENSION => NUCLEAR CHARGE
C
C NOTES:    IF SYMBOL IS NOT RECOGNISED, I.E.NOT IN Z0 RANGE 1 & 'NSYM',
C           THEN THE INTEGER 'IZ0' IS RETURNED AS ZERO.
C
C ROUTINES: NONE
C
C
C AUTHOR:   Martin O'Mullane
C DATE:     25-07-2007
C
C Version : 1.1
C Date    : 25-07-2007
C Modified: Martin O'Mullane
C           - First version.
C
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      CHARACTER*(*)       ELEM
      INTEGER             IZ0
```

9.126 xxeryd: Subroutine xxeryd from library adaslib

```
      SUBROUTINE XXERYD( BWNO , IL , WA  ,
     &                           ER , XIA
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXERYD *********************
C
C  PURPOSE: TO CALCULATE THE ENERGY LEVELS IN RYDBERGS ( FROM WAVE NUM-
C           BERS) RELATIVE TO LEVEL 1, AND THE ENERGIES (ALSO IN RYD.)
C           RELATIVE TO THE IONISATION POTENTIAL.
C
C  CALLING PROGRAM:  GENERAL USE
C
C  SUBROUTINE:
C
C  INPUT :    (I*4)  BWNO    = IONISATION POTENTIAL (CM-1)
C  INPUT :    (I*4)  IL      = NUMBER OF ENERGY LEVELS
C  INPUT :    (R*8)  WA()    = ENERGY RELATIVE TO LEVEL 1 (CM-1)
C
C  OUTPUT:    (R*8)  ER()    = ENERGY RELATIVE TO LEVEL 1 (RYDBERGS)
C  OUTPUT:    (R*8)  XIA()   = ENERGY RELATIVE TO ION. POT. (RYDBERGS)
C
C             (R*8)  WN2RYD  = WAVE NUMBER (CM-1) TO RYDBERG CONVERSION
C                              (PARAMETER)
C
C             (I*4)  I       = GENERAL USE
C
C ROUTINES:  NONE
C
C NOTES:     ION. POT. = IONISATION POTENTIAL
C
C AUTHOR:    PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C            K1/0/81
C            JET EXT. 4569
C
C DATE:      05/07/90
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IL
      REAL*8              BWNO,        ER(IL),      WA(IL),      XIA(IL)
```

9.127 xxfchr: Subroutine xxfchr from library adaslib

```
      SUBROUTINE XXFCHR( CSTRNG , SSTRNG, IFIRST , ILAST )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXFCHR *********************
C
C  PURPOSE: TO IDENTIFY THE FIRST AND LAST OCCURRENCE OF SSTRNG IN
C           CSTRNG, THE VALUES OF WHICH ARE IFIRST , ILAST.
C
C           - IF    NO    OCCURRENCE OF SSTRNG THEN IFIRST=ILAST=0
C           - IF ONLY ONE OCCURRENCE OF SSTRNG THEN IFIRST=ILAST>0
C           - IF OVER ONE OCCURRENCE OF SSTRNG THEN IFIRST>ILAST>0
C
C  NOTE   : ANY TRAILING BLANKS IN THE SEARCH STRING (SSTRNG) ARE
C           IGNORED.
C
C  CALLING PROGRAM: GENERAL USE
C
C  SUBROUTINE:
C
C  INPUT : (C*(*)) CSTRNG   = INPUT STRING FOR INTERROGATION
C  INPUT : (C*(*)) SSTRNG   = INPUT SEARCH STRING
C
C  OUTPUT: (I*4)   IFIRST   = BYTE POSITION OF FIRST OCCURRENCE OF SSTRNG
C                             IN CSTRNG.
C  OUTPUT: (I*4)   ILAST    = BYTE POSITION OF LAST  OCCURRENCE OF SSTRNG
C                             IN CSTRNG.
C
C           (I*4)   ILENC    = LENGTH OF 'CSTRNG' STRING IN BYTES
C           (I*4)   ILENS    = POSITION OF LAST  NON-BLANK BYTE IN SSTRNG
C           (I*4)   I        = GENERAL USE - INCLUDING DUMP FOR UNWANTED
C                             OUTPUT FROM XXSLEN SUBROUTINE.
C
C ROUTINES:
C           ROUTINE    SOURCE    BRIEF DESCRIPTION
C           ------------------------------------------------------------
C           XXSLEN     ADAS      FIND FIRST/LAST NONBLANK BYTES IN STRING
C
C NOTE  :  ANY TRAILING BLANKS IN THE SEARCH STRING (SSTRNG) ARE
C          IGNORED.
C
C
C AUTHOR:  LALIT JALOTA (TESSELLA SUPPORT SERVICES PLC)
C
C DATE  :  27/10/94
C
C UPDATE:  06/03/95 - REVISED BY PAUL BRIDEN (TESSELLA SUPPORT SERVICES)
C                     1) ANALYSE ALL OF CSTRNG (NOT JUST NON-BLANK
C                        PART).
C                     2) ONLY IGNORE TRAILING BLANKS FOR SSTRNG
C                        (KEEP LEADING BLANKS).
C                     3) MODIFY DO LOOP INDEX RANGE TO ENSURE THAT
C                        YOU DO NOT GO BEYOND THE END OF CSTRNG.
C                     4) VERIFY LENGTH OF SSTRNG IS NON-ZERO.
C
C UPDATE:  17/05/07 - Allan Whiteford
C                     Updated comments as part of subroutine
C                     documentation procedure.
C
C-----------------------------------------------------------------------
```

```
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      CHARACTER*(*)       CSTRNG,      SSTRNG
      INTEGER             IFIRST,      ILAST
```

9.128 xxfcse: Subroutine xxfcse from library adaslib

```
      subroutine xxfcse(filein,fileout,type)


C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXFCSE *********************
C
C  PURPOSE: Read in a file, convert it all to upper or lower case and
C           then write it to another file.
C
C  CALLING PROGRAM: GENERAL USE.
C
C  INPUT    : (C*(*)) FILEIN = Input Filename
C  INPUT    : (C*(*)) FILEOUT = Output Filename
C  INPUT    : (C*2)   TYPE = Type of case to convert to:
C                       'UC' -> Convert to Upper Case
C                       'LC' -> Convert to Lower Case
C                       Anything else -> No conversion
C
C  OUTPUT   : NONE
C
C  ROUTINES :
C            ROUTINE    SOURCE    BRIEF DESCRIPTION
C            ------------------------------------------------------------
C            XXCASE     ADAS      CONVERT STRING TO UPPER OF LOWER CASE
C
C  AUTHOR   : Allan Whiteford,
C             University of Strathclyde
C
C  VERSION  : 1.1                          DATE: 05/09/2001
C  MODIFIED : Allan Whiteford
C             First version.
C
C-----------------------------------------------------------------------
      CHARACTER*(*)       FILEIN,      FILEOUT
      CHARACTER*2         TYPE
```

9.129 xxflnm: Subroutine xxflnm from library adaslib

```
      SUBROUTINE XXFLNM( DSNIN  , DSNFUL , LEXIST )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXFLNM *********************
C
C  PURPOSE: TO PREPARE A UNIX DATASET NAME FROM A STRING WHICH MAY
C           INCLUDE AN ADAS ENVIRONMENT LEADER AND COMMENTS.
C           THE ADAS ENVIRONMENT VARIABLE MUST BE FIRST AND IN DOUBLE
C           QUOTES.  THE COMMENTS MUST EITHER FOLLOW OR PRECEDE A COLON.
C
C  CALLING PROGRAM: GENERAL USE
C
C  SUBROUTINE:
C
C  INPUT : (C80)  DSNIN    = INPUT STRING FOR INTERROGATION
C
C  OUTPUT: (C80)  DSNFUL   = THE FULL EXPANDED FILE NAME WITHOUT
C                            EXTRANEOUS MATERIAL
C  OUTPUT: (L*4)  LEXIST   = .TRUE. => NAME FORMED AND FILE EXISTS
C                            .FALSE.=> FAILED TO FORM NAME OR FIND FILE
C
C          (C*80) DSN1     = WORK STRING
C          (C*80) DSNTEMP  = WORK STRING
C          (C*80) BLANK    = BLANK STRING
C          (I*4)   LEN1     = STRING INDEX
C          (I*4)   LEN2     = STRING INDEX
C          (I*4)   LEN3     = STRING INDEX
C          (I*4)   LEN4     = STRING INDEX
C
C ROUTINES:
C          ROUTINE    SOURCE   BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          XXSLEN     ADAS     FIND BEGINNING AND END OF A STRING
C          I4UNIT     ADAS     FETCH UNIT NUMBER FOR OUTPUT OF MESSAGES
C          GETENV     UNIX     FETCH AN ENVIRONMENT VARIABLE
C
C
C AUTHOR:  H. P. SUMMERS, UNIVERSITY OF STRATHCLYDE
C          JA8.08
C          TEL. 0141-553-4196
C
C DATE  :  21/05/96
C
C UPDATE:
C
C VERSION: 1.1                          DATE: 29-05-96
C MODIFIED: WILLIAM OSBORN
C               - ADDED CODE FOR INITIAL COMMENTS AND PUT INTO S.C.C.S.
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
C

      CHARACTER*80        DSNFUL,      DSNIN
      LOGICAL             LEXIST
```

9.130 xxflsh: Subroutine xxflsh from library adaslib

```
      SUBROUTINE XXFLSH(PIPEOU)
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXFLSH *********************
C
C  PURPOSE: ROUTINE FOR SETTING CALL TO "FLUSH" COMMAND DEPENDING ON
C     OPERATING SYSTEM SPCIFICS.
C
C  CALLING PROGRAM: ANY ADAS PROGRAM WHICH COMMUNICATES VIA A PIPE
C     PARTICULARLY TO IDL INTERFACE ROUTINES.
C
C
C  SUBROUTINE:
C
C          (I*4)  PIPEOU  = UNIT NUMBER OF PIPE USED
C
C NOTE:
C
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C
C
C AUTHOR:  L. JALOTA (TESSELLA SUPPORT SERVICES PLC)
C
C DATE:     6/03/95
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      INTEGER             PIPEOU
```

9.131 xxfrmt_t: Subroutine xxfrmt_t from library adaslib

```
       subroutine xxfrmt_t( ita_max , isa_max , ila_max , xja_max ,
     &                      iw      , fmt02
     &                    )
c-----------------------------------------------------------------------
c
c ****************** fortran 77 subroutine: xxfrmt_t ******************
c
c  purpose: To determine the format and file length required for the
c           transition strings of an ion
c
c
c subroutine:
c
c  input : (i*4)  ita_max   = largest value for transition index
c  input : (i*4)  isa_max   = largest value for transition multiplicity
c  input : (i*4)  ila_max   = largest value for transition orb. ang.mom
c  input : (r*8)  xja_max   = largest value for transition (stat.wt-1)/2
c
c  output: (i*4)  iw        = field length required for trans. strings
c  output: (c*63) fmt02     = appropriate format specification
c
c routines:
c          routine    source    brief description
c          ------------------------------------------------------
c         i4unit     adas      fetch unit number for output of messages
c
c
c author:  h. p. summers, university of strathclyde
c          ja7.08
c          tel. 0141-548-4196
c
c date:    26/07/06
c
c version: 1.1                          date: 26/07/2006
c modified: hugh p summers
c               - first edition.
c
c-----------------------------------------------------------------------
      CHARACTER*63        FMT02
      INTEGER             ILA_MAX,     ISA_MAX,     ITA_MAX,     IW
      REAL*8              XJA_MAX
```

9.132 xxfrmt_trm: Subroutine xxfrmt_trm from library adaslib

```
       subroutine xxfrmt_trm( isa_max , ila_max , xja_max ,
     &                        iw      , fmt03
     &                    )
c-----------------------------------------------------------------------
c
c ****************** fortran 77 subroutine: xxfrmt_trm *****************
c
c  purpose: To determine the format and string length required for the
c           term strings of an ion
c
c
c subroutine:
c
c  input : (i*4)  isa_max   = largest value for term multiplicity
c  input : (i*4)  ila_max   = largest value for term orb. ang.mom
c  input : (r*8)  xja_max   = largest value for term (stat.wt-1)/2
c
c  output: (i*4)  iw        = field length required for term strings
c  output: (c*63) fmt03     = appropriate format specification
c
c routines:
c          routine    source    brief description
c          ------------------------------------------------------------
c         i4unit     adas      fetch unit number for output of messages
c
c
c author:  h. p. summers, university of strathclyde
c          ja7.08
c          tel. 0141-548-4196
c
c date:    26/07/06
c
c version: 1.1                          date: 26/07/2006
c modified: hugh p summers
c               - first edition.
c
c-----------------------------------------------------------------------
      CHARACTER*31        FMT03
      INTEGER             ILA_MAX,     ISA_MAX,     IW
      REAL*8              XJA_MAX
```

9.133 xxgama: Subroutine xxgama from library adaslib

```
      SUBROUTINE XXGAMA( MXINDX , JGAM, GAM )
C
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXGAMA *********************
C
C  PURPOSE: SETUP LOOKUP TABLES FOR ROUTINES I4JGAM AND R8GAM
C
C           TABLES 'JGAM' AND 'GAM' CAN BE REFERENCED WITH FUNCTIONS
C           'I4JGAM' AND 'R8GAM', RESPECTIVELY.
C
C  CALLING PROGRAM: GENERAL USE.
C
C  INPUT : (I*4)  MXINDX  =
C
C  OUTPUT: (I*4)  JGAM()  =
C                           DIMENSION: REFERENCED BY I.
C  OUTPUT: (I*4)  GAM()   =
C                           DIMENSION: REFERENCED BY I.
C
C  PARAM : (R*8)  P1      = 64.0
C
C          (I*4)  I       = LOOP INDEX.
C          (I*4)  I1      = I-1
C          (I*4)  J1      = USED IN COMPUTATION ( = JGAM(N) ).
C
C          (R*8)  X1      = USED IN COMPUTATION ( = GAM(N) ).
C
C ROUTINES: NONE
C
C AUTHOR:   JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C           K1/0/81
C           JET EXT. 5183
C
C DATE:     30/09/93
C
C VERSION  : 1.2
C DATE     : 20-12-2001
C MODIFIED : Martin O'Mullane
C               - Removed mainframe listing information beyond column 72.
C
C VERSION  : 1.3
C DATE     : 10-04-2007
C MODIFIED : Allan Whiteford
C               - Modified documentation as part of automated
C   subroutine documentation preparation.
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             JGAM(MXINDX),             MXINDX
      REAL*8              GAM(MXINDX)
```

9.134 xxgtsl: Subroutine xxgtsl from library adaslib

```
       SUBROUTINE XXGTSL( N  , D  , DU , DL , B  , IFAIL )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXGTSL *********************
C
C  PURPOSE: GIVEN A GENERAL TRIDIAGONAL MATRIX AND A RIGHT HAND SIDE
C           WILL FIND THE SOLUTION OF THE ASSOCIATED SYSTEM OF LINEAR
C           EQUATIONS.
C
C  CALLING PROGRAM: GENERAL USE
C
C  SUBROUTINE:
C  INPUT :    (I*4)  N       = ORDER OF TRIDIAGONAL MATRIX
C
C  INPUT :    (R*8)  DL()    = SUBDIAGONAL OF THE MATRIX.  DL(2) THROUGH
C                              DL(N) SHOULD CONTAIN THE SUBDIAGONAL. ON
C                              OUTPUT DL IS DESTROYED
C  INPUT :    (R*8)  D()     = DIAGONAL OF THE MATRIX.  ON OUTPUT D
C                              IS DESTROYED
C  INPUT :    (R*8)  DU()    = SUPERDIAGONAL OF THE MATRIX. DU(2) THROUGH
C                              DU(N) SHOULD CONTAIN THE SUPERDIAGONAL.
C                              OUTPUT DU IS DESTROYED
C  INPUT :    (R*8)  B()     = RIGHT HAND SIDE VECTOR
C
C  OUTPUT:    (R*8)  B()     = SOLUTION VECTOR
C
C  OUTPUT:    (I*4)  IFAIL   = 0 - NORMAL VALUE
C                              = K - IF THE KTH PIVOT ELEMENT BECOMES
C                                    BECOMES EXACTLY ZERO.  THE ROUTINE
C                                    RETURNS WHEN THIS IS THE CASE.
C
C             (I*4)  K       = GENERAL INTEGER
C             (I*4)  KB      = GENERAL INTEGER
C             (I*4)  KP1     = K+1
C             (I*4)  NM1     = N-1
C             (I*4)  NM2     = N-2
C             (R*8)  T       = GENERAL REAL
C
C ROUTINES:  NONE
C
C NOTE:
C            TRANSCRIBED FROM LINPACK PUBLICATION.  VERSION DATED
C            08/14/78, JACK DONGARRA, ARGONNE NATIONAL LABORATORY
C
C AUTHOR:  H. P. SUMMERS, UNIVERSITY OF STRATHCLYDE
C          JA8.08
C          TEL. 0141-553-4196
C
C DATE:    04/07/95
C
C DATE:    10/07/95 VERSION 1.1
C UPDATE:  TIM HAMMOND, TESSELLA SUPPORT SERVICES PLC
C    - UNIX PORT
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IFAIL,       N
      REAL*8              B(N),        D(N),        DL(N),       DU(N)
```

9.135 xxguid: Subroutine xxguid from library adaslib

```
      SUBROUTINE XXGUID( USERID )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXGUID *********************
C
C  PURPOSE: TO FETCH USER IDENTIFIER FROM UNIX
C
C  CALLING PROGRAM:
C
C  SUBROUTINE:
C
C  OUTPUT: (C*6)   USERID   = USER IDENTIFIER
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C
C
C UNIX-IDL PORT:
C    WILLIAM OSBORN, TESSELLA SUPPORT SERVICES PLC.
C
C DATE:    22ND APRIL 1996
C
C VERSION: 1.1 DATE: 22-04-96
C MODIFIED: WILLIAM OSBORN
C     - FIRST VERSION.
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      CHARACTER*10        USERID
```

9.136 xxhkey: Subroutine xxhkey from library adaslib

```
      SUBROUTINE XXHKEY( CTEXT  , CKEY , CBREAK , CANS )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXHKEY *********************
C
C  PURPOSE: TO EXTRACT FROM A LINE OF TEXT 'CTEXT' A RESPONSE TO A KEY
C           IN THER FORM OF '<CKEY> = <CANS>'.
C
C  CALLING PROGRAM: GENERAL USE
C
C  SUBROUTINE:
C
C  INPUT : (C*(*)) CTEXT   = INPUT TEXT LINE CONTAINING KEY & RESPONSES
C  INPUT : (C*(*)) CKEY    = KEY TEXT
C  INPUT : (C*1  ) CBREAK  = KEY/RESPONSE PAIR SEPERATOR SYMBOL
C
C  OUTPUT: (C*(*)) CANS    = RERSPONSE FOR GIVEN KEY: BLANK IF NOT FOUND
C
C          (I*4)   LENTXT  = LENGTH IN BYTES OF 'CTEXT' STRING
C          (I*4)   LENKEY  = LENGTH IN BYTES OF 'CKEY' STRING
C          (I*4)   LENANS  = LENGTH IN BYTES OF 'CANS' STRING
C          (I*4)   IKEY    = LENGTH IN BYTES OF 'CKEY' IGNORING TRAILING
C                            BLANKS
C          (I*4)   IPOS1   = USED IN IDENTIFYING RELEVANT BYTES IN CTEXT
C          (I*4)   IPOS2   = USED IN IDENTIFYING RELEVANT BYTES IN CTEXT
C          (I*4)   IPOS3   = USED IN IDENTIFYING RELEVANT BYTES IN CTEXT
C          (I*4)   I       = GENERAL USE INDEX
C
C ROUTINES: NONE
C
C NOTES:    THIS ROUTINE EXTRACTS FROM 'CTEXT' A RESPONSE TO A GIVEN KEY
C           IN THER FORM OF '<CKEY> = <CANS>'. E.G. 'FILE = DSN001'
C           WOULD REQUIRE AS INPUT CKEY='FILE' AND WOULD GIVE AS OUTPUT
C           CANS='DSN001'. ALL KEY/RESPONSE PAIRS MUST BE SEPARATED BY
C           THE CHARACTER GIVEN BY 'CBREAK' E.G. A SLASH, AND EACH KEY
C           MUST BE FOLLOWED BY AN EQUALS SIGN. THE NUMBER OF SPACES
C           BETWEEN THE KEY AND THE EQUAL SIGN AND BETWEEN THE RESPONSE
C           AND THE EQUAL SIGN IS NOT IMPORTANT.
C
C           THE BYTE PRECEEDING THE KEY MUST BE A BLANK OR 'CBREAK'
C           CHARACTER UNLESS IT STARTS AT BYTE ONE IN 'CTEXT'.
C
C           IF A KEY DOES NOT EXIST IN 'CTEXT' THEN 'CANS' IS RETURNED
C           BLANK.
C
C           THE KEY IS TAKEN AS 'CKEY' REMOVING ANY TRAILING BLANKS.
C           LEADING BLANKS ARE LEFT IN PLACE AND WILL USED WHEN THE
C           THE SEARCH FOR THE KEY IS MADE:
C
C           I.E.  'DATA   ' AND 'DATA' ARE THE SAME KEY BUT
C                 ' DATA '  AND 'DATA ' ARE DIFFERENT KEYS ALTHOUGH
C                 BOTH WILL GIVE THE SAME RESULTS IF A SPACE EXISTS
C                 BEFORE 'DATA' IN THE INPUT TEXT LINE.
C
C           AN EXAMPLE OF AN INPUT TEXT LINE IS:
C
C           8524.0 A    5 7 /FILMEM = FBBH91BE/   CODE=   V2B DLN1   /
C
C           THIS WOULD GIVE THE FOLLOWING:
```

```
C
C           CKEY='FILMEM'  =>  CANS='FBBH91BE'
C           CKEY=' FILMEM' =>  CANS=' '
C           CKEY='CODE'    =>  CANS='V2B DLN1'
C           CKEY=' CODE'   =>  CANS='V2B DLN1'
C           CKEY='OTHER'   =>  CANS=' '
C
C           (IF THE CHARACTER STRING IS SHORTER THAN THE RESPONSE THEN
C            THE RESPONSE IS TRUNCATED ACCORDINGLY.)
C
C           SPACES CAN EXIST IN THE KEY. I.E. CKEY='PLOT A'. BUT CARE
C           SHOULD BE TAKEN WHEN USING PREFIXES ON A COMMON KEY BASE,
C           I.E. 'A PLOT', 'B PLOT'. THIS IS BECAUSE IF A SUBSEQUENT
C           KEY TO BE FOUND IS 'PLOT' THEN  EITHER OF THESE SATISFY
C           THIS CRITERION AS WELL AS 'PLOT' ITSELF.
C
C           AN EXAMPLE OF AN INPUT TEXT LINE IS:
C
C           A FILE=TEST0/B FILE = TEST1/FILE=TEST2/FILE 1=TEST3/FILE 2=/
C
C           THIS WOULD GIVE THE FOLLOWING:
C
C           CKEY='A FILE'  =>  CANS='TEST0'
C           CKEY='B FILE'  =>  CANS='TEST1'
C           CKEY='FILE'    =>  CANS='TEST0' (WRONG RESPONSE PICKED UP)
C           CKEY='FILE 1'  =>  CANS='TEST3'
C           CKEY='FILE 2'  =>  CANS='TEST4'
C
C           IT IS ALSO POSSIBLE TO IMBED RESPONSES
C
C           AN EXAMPLE OF AN INPUT TEXT LINE IS:
C
C           FILE 1 =  Z1 = 23 / FILE = FILE 1 = 6 /
C
C           THIS WOULD GIVE THE FOLLOWING:
C
C           CKEY='FILE 1'  =>  CANS='Z1 = 23'
C           CKEY=' FILE 1' =>  CANS='6'
C           CKEY='Z1'      =>  CANS='23'
C           CKEY='FILE'    =>  CANS='FILE 1 = 6'
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/37
C          JET EXT. 2520
C
C DATE:    26/04/91
C
C VERSION  : 1.2
C DATE     : 31-05-2007
C MODIFIED : H P Summers
C               - Increased robustness for single letter keys occurring
C                 elsewhere at non-key positions in strings.
C
C-----------------------------------------------------------------------
      CHARACTER*(*)       CANS
      CHARACTER           CBREAK
      CHARACTER*(*)       CKEY,        CTEXT
```

9.137 xxhunt: Subroutine xxhunt from library adaslib

```
      SUBROUTINE XXHUNT( XX , N , X , JLO )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXHUNT *********************
C
C  PURPOSE: GIVEN AN ARRAY XX(1:N), AND GIVEN A VALUE X, RETURNS A
C           VALUE JLO SUCH THAT X IS BETWEEN XX(JLO) AND XX(JLO+1).
C
C           XX(1:N) MUST BE MONOTONIC, EITHER INCREASING OR
C           DECREASING. JLO=0 OR JLO=N IS RETURNED TO INDICATE THAT
C           X IS OUT OF RANGE. JLO ON INPUT IS TAKEN AS THE INITIAL
C           GUESS FOR JLO ON OUTPUT.
C
C  REFERENCE:  NUMERICAL RECIPES: The Art of Scientific Computing
C              (FORTRAN Version).
C              W.H.Press, B.P.Flannery, S.A.Teukolsky & W.T.Vetterling.
C              (Cambridge University Press, Cambridge). 1989. p.233
C              ISBN 0 521 38330 7
C
C  CALLING PROGRAM: GENERAL USE
C
C  SUBROUTINE:
C
C  INPUT :    (R*8)  XX      = MONOTONIC INPUT ARRAY
C  INPUT :    (I*4)  N       = INPUT ARRAY 'XX()' LENGTH
C  INPUT :    (R*8)  X       = VALUE WHERE INTERVAL IS TO BE FOUND
C  I/O   :    (I*4)  JLO     = INPUT: INITIAL GUESS FOR INTERVAL INDEX
C                              OUTPU: RETURNED INTERVAL INDEX
C
C             (I*4)  INC     = HUNT INCREMENT
C             (I*4)  JHI     = UPPER BOUND FOR BRACKETING X
C             (I*4)  JM      = MEAN OF JLO AND JHI
C
C             (L*4)  ASCND   = .TRUE. FOR ASCENDING VALUES IN XX,
C                              .FALSE. FOR DESCENDING VALUES
C
C ROUTINES:  NONE
C
C NOTE:
C
C AUTHOR:  LORNE D. HORTON (IPP GARCHING)
C          L5.213
C          IPP EXT. 1635
C
C DATE:    19/03/03
C
C VERSION:  1.1                          DATE: 19/03/03
C MODIFIED: Lorne Horton
C           - Initial version
C
C UPDATE:   1.2                          DATE: 17/05/07
C MODIFIED: Allan Whiteford
C           - Updated comments as part of subroutine documentation
C             procedure.
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             JLO,         N
      REAL*8              X,           XX(N)
```

9.138 xxi4ss: Subroutine xxi4ss from library adaslib

```
      subroutine xxi4ss(n, lup, ia, itag)

C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXI4SS ********************
C
C  PURPOSE: Sorts an integer array xa and its index array. This is a
C           bubble sort designed for small arrays.
C
C
C  CALLING PROGRAM:  General use
C
C
C  SUBROUTINE:
C
C  INPUT       :  (I*4)  N     = Size of input arrays
C  INPUT       :  (L*4)  LUP   = .TRUE. sort in ascending order
C                                .FALSE. descending order
C  INPUT/OUPUT :  (I*4)  IA    = Array to be sorted
C  INPUT/OUPUT :  (I*4)  ITAG  = Original index of sorted XA
C
C  ROUTINES    : NONE
C
C  NOTES       : Shell sort from Numerical Receipies.
C
C  AUTHOR      : Martin O'Mullane
C
C  DATE        : 7-02-2000
C
C  VERSION     : 1.1
C
C
C-----------------------------------------------------------------------
      INTEGER             IA(*),       ITAG(*),     N
      LOGICAL             LUP
```

9.139 xxidtl: Subroutine xxidtl from library adaslib

```
       SUBROUTINE XXIDTL( INDEX , N , L )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXIDTL *********************
C
C  PURPOSE:  INVERSE OF FUNCTION I4IDFL. RETURNS THE UNIQUE N AND L
C            QUANTUM NUMBERS WHICH GENERATE THE GIVEN INDEX WHEN PASSED
C            TO I4IDFL.
C
C  CALLING PROGRAM: GENERAL USE.
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  INDEX    = INDEX NUMBER.
C
C  OUTPUT: (I*4)  N        = N QUANTUM NUMBER.
C  OUTPUT: (I*4)  L        = L QUANTUM NUMBER.
C
C           (I*4)  ID       = INDEX RETURNED BY FUNCTION I4IDFL
C
C ROUTINES:
C           ROUTINE    SOURCE    BRIEF DESCRIPTION
C           ------------------------------------------------------------
C           I4IDFL     ADAS      RETURNS UNIQUE INDEX FROM QUANTUM
C                                NUMBERS N AND L.
C
C AUTHOR:  JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C           K1/0/81
C           JET EXT. 5183
C
C DATE:    10/09/93
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      INTEGER             INDEX,       L,           N
```

9.140 xxidtm: Subroutine xxidtm from library adaslib

```
       SUBROUTINE XXIDTM( INDEX , N , L , M )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXIDTM *********************
C
C  PURPOSE:  INVERSE OF FUNCTION I4IDFM. RETURNS THE UNIQUE N, L AND M
C            QUANTUM NUMBERS WHICH GENERATE THE GIVEN INDEX WHEN PASSED
C            TO I4IDFM.
C
C  CALLING PROGRAM: GENERAL USE.
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  INDEX   = INDEX NUMBER.
C
C  OUTPUT: (I*4)  N       = N QUANTUM NUMBER.
C  OUTPUT: (I*4)  L       = L QUANTUM NUMBER.
C  OUTPUT: (I*4)  M       = M QUANTUM NUMBER.
C
C          (I*4)  ID      = INDEX RETURNED BY FUNCTION I4IDFM
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4IDFM     ADAS      RETURNS UNIQUE INDEX FROM QUANTUM
C                               NUMBERS N, L AND M.
C
C AUTHOR:  JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 5183
C
C DATE:    10/09/93
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      INTEGER             INDEX,       L,           M,           N
```

9.141 xxin17: Subroutine xxin17 from library adaslib

```
      SUBROUTINE XXIN17( IUNIT  , ICLASS , DSNAME , LERROR ,
     &                   NDDEN  , NDTIN  , NDZ1V  ,
     &                   IPRTD  , ISYSD  ,
     &                   IDE    , ITE    , IZE    ,
     &                   DENSR  , TR     , ZIPT   ,
     &                   LSWIT  , EIA    ,
     &                   AIPT
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXIN17 *********************
C
C  PURPOSE: TO OPEN AND ACQUIRE DATA FROM MASTER CONDENSED
C           COLLISIONAL-DIELECTRONIC FILES:
C
C           THE FOLLOWING FILES ARE ALLOWED:
C
C             1. RECOMBINATION COEFFICIENTS
C             2. IONISATION    COEFFICIENTS
C             3. CHARGE-EXCHANGE RECOMBINATION COEFFICIENTS
C             4. METASTABLE CROSS-COUPLING COEFFICIENTS
C             5. PARENT METASTABLE CROSS-COUPLING COEFFICIENTS
C             6. RECOMBINATION-BREMSSTRAHLUNG  POWER COEFFICIENTS
C             7. CHARGE-EXCHANGE RECOMBINATION POWER COEFFICIENTS
C
C             (NOTE: SPECIFIC AND TOTAL LOW LINE POWER COEFFICIENTS
C                    SHOULD BE READ USING 'XXIN80'.
C                    IF  ONLY STANDARD FILES ARE TO BE READ BY  THE
C                    PROGRAM USE 'XXINST'.)
C
C  CALLING PROGRAM: GENERAL USE
C
C  DATA:
C           THE  SOURCE  DATA IS CONTAINED AS  MEMBERS  OF  PARTITIONED
C           DATA SETS AS FOLLOWS:
C
C             1. JETUID.ACD<YR>.DATA
C             2. JETUID.SCD<YR>.DATA
C             3. JETUID.CCD<YR>.DATA
C             4. JETUID.QCD<YR>.DATA
C             5. JETUID.XCD<YR>.DATA
C             6. JETUID.PRB<YR>.DATA
C             7. JETUID.PRC<YR>.DATA
C
C           WHERE <YR> DENOTES TWO INTEGERS FOR THE YEAR SELECTED.
C           IF <YR> IS BLANK THEN THE CURRENT RECOMMENDED DATA SETS ARE
C           USED
C
C           THE MEMBERS OF THE PARTITIONED DATA SETS ARE EITHER:
C           1)  <SE><I><J>  FOR PARTIAL MASTER CONDENSED FILES, OR
C           2)  <SE>        FOR STANDARD MASTER CONDENSED FILES
C
C           WHERE: <SE> IS THE ONE OR TWO LETTER ION SEQUENCE CODE.
C                  <I>  IS A  SINGLE  INTEGER  REPRESENTING  THE  PARENT
C                       INDEX OR METASTABLE INDEX DEPENDING ON THE DATA
C                       SET CLASS AND PRODUCTION BY BUNDLE-NS OR
C                       LOW-LEVEL+PROJECTION MODELS
C                  <J>  IS A SINGLE INTEGER REPRESENTING THE SPIN SYSTEM
C                       INDEX, METASTABLE INDEX OR PARENT INDEX
```

```
C                       DEPENDING ON THE DATA SET CLASS AND PRODUCTION
C                       BY BUNDLE-NS OR LOW-LEVEL+PROJECTION MODELS
C
C                  E.G. PARTIAL  FILES: 'C12' OR 'HE21'
C                       STANDARD FILES: 'C'   OR 'HE'
C
C           THE  'PARTIAL'  AND  'STANDARD'  MASTER CONDENSED FILES ARE
C           IDENTICAL IN FORM,   EXCEPT  THAT  THREE  ADDITIONAL  LINES
C           ARE INCLUDED  AT  THE  BEGINNING OF  THE  'PARTIAL'  MASTER
C           FILES.   THE  FIRST  OF THESE LINES CONTAINS  A ROW OF  '='
C           SIGNS,  THE SECOND A PARENT/SPIN (OR EQUIVALENTS) PARAMETER
C           LIST,   AND  THE
C           THIRD A ROW OF "-" SIGNS. THIS DIFFERENCE IS USED TO IDENT-
C           IFY WHICH FILE TYPE IS BEING READ.
C
C           THE CHARACTER STRING SEPARATING THE INPUT DATA FOR EACH
C           VALUE OF Z1 IN THE FILE WILL GIVE:
C
C           PARTIAL & STANDARD: THE Z1 VALUE (Z1=) AND DATE (DATE:).
C                (OLDER DATA SETS MAY HAVE 'Z =' INSTEAD OF 'Z1=' HERE)
C           PARTIAL FILES ONLY: THE PARENT (IPRT=) & SPIN SYSTEM (ISYS=)
C                OR EQUIVALENTS (IGRD=) & (IGRD=, JGRD= AND JPRT=)
C                AS FOLLOW:-
C
C               ICLASS     INDI     INDJ
C               ------     ----     ----
C                 1        IPRT     IGRD (OR ISYS)
C                 2        IPRT     IGRD (OR ISYS)
C                 3        IPRT     IGRD (OR ISYS)
C                 4        IPRT     IGRD (OR ISYS)
C                 5        IPRT     IGRD (OR ISYS)
C                 6        IGRD     JGRD
C                 7        IPRT     JPRT
C
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  IUNIT   = UNIT TO WHICH INPUT DATA SET ALLOCATED
C  INPUT : (I*4)  ICLASS  = UNIT TO WHICH INPUT DATA SET ALLOCATED
C  INPUT : (C*(*))DSNAME  = INPUT MASTER CONDENSED FILE DATA SET NAME
C  OUTPUT: (L*4)  LERROR  = .TRUE.  => ERROR DETECTED IN READING FILE
C                         = .FALSE. => NO ERROR DETECTED IN FILE
C
C  INPUT : (I*4)  NDDEN   = MAX. NUMBER OF REDUCED DENSITIES ALLOWED IN
C                           MASTER CONDENSED FILE FOR A GIVEN SEQUENCE
C  INPUT : (I*4)  NDTIN   = MAX. NO. OF REDUCED TEMPERATURES ALLOWED IN
C                           MASTER CONDENSED FILE FOR A GIVEN SEQUENCE
C  INPUT : (I*4)  NDZ1V   = MAX. NUMBER OF CHARGE STATES ALLOWED IN
C                           MASTER CONDENSED FILE FOR A GIVEN SEQUENCE
C
C  INPUT : (I*4)  IPRTD   = INPUT PARTIAL MASTER CONDENSED FILE:
C                           PARENT INDEX SPECIFIED IN DATA SET NAME.
C  INPUT : (I*4)  ISYSD   = INPUT PARTIAL MASTER CONDENSED FILE: SPIN-
C                           SYSTEM INDEX SPECIFIED IN DATA SET NAME.
C
C  OUTPUT: (I*4)  IDE     = NUMBER OF REDUCED DENSITIES READ FROM INPUT
C                           MASTER CONDENSED FOR A GIVEN SEQUENCE
C  OUTPUT: (I*4)  ITE     = NO. OF REDUCED TEMPERATURES READ FROM INPUT
C                           MASTER CONDENSED FOR A GIVEN SEQUENCE
C  OUTPUT: (I*4)  IZE     = NO. OF CHARGE STATES GIVEN IN THE INPUT
C                           MASTER CONDENSED FOR A GIVEN SEQUENCE
```

```
C
C  OUTPUT: (R*8)  DENSR() = SET OF 'IDE' INPUT REDUCED DENSITIES (CM-3/
C                           Z1**7) READ FROM CONDENSED MASTER FILE.
C  OUTPUT: (R*8)  TR()    = SET OF 'ITE'  INPUT  REDUCED  TEMPERATURES
C                           (K/Z1**2) READ FROM CONDENSED MASTER FILE.
C  OUTPUT: (R*8)  ZIPT()  = SET OF 'IZE' INPUT CHARGE STATES READ FROM
C                           CONDENSED MASTER FILE.
C                           (CHARGE STATE = RECOMBINING ION CHARGE)
C
C  OUTPUT: (L*4)  LSWIT   = .TRUE.  => IONISATION POTENTIALS
C                                      INCLUDED IN INPUT MASTER FILE.
C                           .FALSE. => IONISATION POTENTIALS
C                                      NOT INCLUDED IN INPUT MASTER FILE
C  OUTPUT: (R*8)  EIA()   = IONISATION POTENTIALS: ()=ION CHARGE
C                           UNITS: WAVE NUMBERS (CM-1)
C                           (= 0.0 IF NOT SET)
C
C  OUTPUT: (R*8)  AIPT(,,)= CONDENSED MASTER FILE DATA. COLL-DIEL COEFF.
C                           1ST DIMENSION: REDUCED DENSITY ('DENSR()')
C                           2ND DIMENSION: REDUCED TEMPERATURE ('TR()')
C                           3RD DIMENSION: CHARGE STATE ('ZIPT()')
C
C          (I*4)  I4UNIT  = FUNCTION (SEE ROUTINE SECTION BELOW)
C          (I*4)  IPRT    = INPUT PARTIAL MASTER CONDENSED FILE:
C                           PARENT INDEX READ FROM INPUT FILE.
C          (I*4)  ISYS    = INPUT PARTIAL MASTER CONDENSED FILE:
C                           SPIN-SYSTEM INDEX READ FROM INPUT FILE.
C          (I*4)  IPOT    = NUMBER OF IONISATION POTENTIAL VALUES
C                           PRESENT IN THE INPUT FILE.
C          (I*4)  IZ1     = CHARGE STATE READ FROM THE LINE PRECEEDING
C                           AN INPUT BLOCK FROM THE FILE.
C                           (= RECOMBINING ION CHARGE)
C          (I*4)  IBGN    = FIRST BYTE OF INTEREST IN CHARACTER 'STRING'
C          (I*4)  IEND    = LAST  BYTE OF INTEREST IN CHARACTER 'STRING'
C          (I*4)  ID      = ARRAY SUBSCRIPT USED FOR DENSITY VALUES
C          (I*4)  IT      = ARRAY SUBSCRIPT USED FOR TEMPERATURE VALUES
C          (I*4)  IZ      = ARRAY SUBSCRIPT USED FOR ION-CHARGE VALUES
C          (I*4)  I       = GENERAL USE
C
C          (L*4)  LPART   = .TRUE.  => REQUESTED INPUT FILE: PARTIAL
C                         = .FALSE. => REQUESTED INPUT FILE: STANDARD
C
C          (C*5)  CPOT    = 'IPOT'
C          (C*5)  CHINDI  = 'IPRT= ' OR 'IGRD= ' DEPENDING ON ICLASS
C          (C*5)  CHINDJ  = 'IGRD= ', 'JGRD= ' OR 'JPRT= ' DEPENDING
C                            ON ICLASS
C          (C*80) STRING  = STRING INTO WHICH 1ST LINE OF INPUT FILE IS
C                           READ TO ENABLE ITS FORMAT TO BE ESTABLISHED.
C
C NOTE:
C          STREAM HANDLING:
C             STREAM 'IUNIT' IS USED FOR READING CONDENSED MASTER FILES
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4UNIT     ADAS      FETCH UNIT NUMBER FOR OUTPUT OF MESSAGES
C          XXREIA     ADAS      READ IN UNKNOWN NUMBER OF 'EIA' VALUES
C                               IF PRESENT.
C
C
```

```
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 4569
C
C DATE:    22/08/90
C
C UPDATE:  05/03/91 - PE BRIDEN - ADAS91: REMOVED OPENING OF DATA SET
C
C UPDATE:  23/04/93 - PE BRIDEN - ADAS91: ADDED I4UNIT FUNCTION TO WRITE
C                                         STATEMENTS FOR SCREEN MESSAGES
C
C UPDATE:  24/05/93 - PE BRIDEN - ADAS91: CHANGED I4UNIT(0)-> I4UNIT(-1)
C
C UPDATE:  11/08/93 - HP SUMMERS - CHANGED TO ACCEPT EXTRA DATA CLASSES
C                                  AND USE OF IGRD, JGRD, IPRT, JPRT AS
C                                  ALTERNATIVES TO IPRT AND ISYS.
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 06-09-95
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST RELEASE
C
C VERSION  : 1.2
C DATE     : 10-04-2007
C MODIFIED : Allan Whiteford
C               - Modified documentation as part of automated
C   subroutine documentation preparation.
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      CHARACTER*(*)       DSNAME
      INTEGER             ICLASS,      IDE,         IPRTD,       ISYSD
      INTEGER             ITE,         IUNIT,       IZE,         NDDEN
      INTEGER             NDTIN,       NDZ1V
      LOGICAL             LERROR,      LSWIT
      REAL*8              AIPT(NDDEN,NDTIN,NDZ1V),  DENSR(NDDEN)
      REAL*8              EIA(250),    TR(NDTIN),   ZIPT(NDZ1V)
```

9.142 xxin80: Subroutine xxin80 from library adaslib

```
      SUBROUTINE XXIN80( IUNIT  , DSNAME , LERROR ,
     &                   NDDEN  , NDTIN  , NDZ1V  , NDMET ,
     &                   IDE    , ITE    , IZE    ,
     &                   DENSR  , TR     , ZIPT   ,
     &                   IME    , IMETR  , CSTRGA ,
     &                   NPRNT  , IPRNT  , IPSYS  ,
     &                   LSWIT  , EIA    ,
     &                   AIPT
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXIN80 *********************
C
C  PURPOSE: TO OPEN AND ACQUIRE DATA FROM MASTER CONDENSED
C           COLLISIONAL-DIELECTRONIC FILES:
C
C           THE FOLLOWING FILES ARE ALLOWED:
C
C             8. TOTAL    LINE POWER COEFFICIENTS
C             9. SPECIFIC LINE POWER COEFFICIENTS
C
C           AND TO OPEN AND ACQUIRE DATA FROM THE FOLLOWING FILE:
C
C            10. METASTABLE POPULATION DATA
C
C             (NOTE: OTHER MASTER  CONDENSED  COLL.-DIEL.  COEFFICIENTS
C                    SHOULD BE READ USING 'XXIN17'.
C                    IF  ONLY STANDARD FILES ARE  TO  BE  READ  BY  THE
C                    PROGRAM USE 'XXINST'.)
C
C  CALLING PROGRAM: GENERAL USE
C
C  DATA:
C           THE  SOURCE  DATA IS CONTAINED AS  MEMBERS  OF  PARTITIONED
C           DATA SETS AS FOLLOWS:
C
C             8. JETUID.PLT<YR>.DATA
C             9. JETUID.PLS<YR>.DATA
C            10. JETUID.MET<YR>.DATA
C
C           WHERE <YR> DENOTES TWO INTEGERS FOR THE YEAR SELECTED.
C           IF <YR> IS BLANK THEN THE CURRENT RECOMMENDED DATA SETS ARE
C           USED
C
C           THE MEMBERS OF THE PARTITIONED DATA SETS ARE EITHER:
C           1)  <SE><#><#>  FOR PARTIAL MASTER CONDENSED FILES, OR
C           2)  <SE>        FOR STANDARD MASTER CONDENSED FILES AND
C                           METASTABLE POPULATION FILE
C
C           WHERE: <SE> IS THE ONE OR TWO LETTER ION SEQUENCE CODE.
C                       NOTE: FOR THE BARE NUCLEUS <SE> = '@'
C                  <#>  IS THE SINGLE CHARACTER '#'
C
C
C                  E.G. PARTIAL  MASTER CONDENSED FILES:'@##' OR 'HE##'
C                       STANDARD AND METASTABLE FILES  : '@'  OR 'HE'
C
C           THE  'PARTIAL'  AND  'STANDARD'  MASTER CONDENSED FILES ARE
C           IDENTICAL IN FORM  EXCEPT:
```

```
C           A) EIGHT  ADDITIONAL  LINES ARE INCLUDED  AT THE  BEGINNING
C              OF  THE  'PARTIAL'  MASTER  FILES.   THE FIRST  OF THESE
C              LINES CONTAINS  A ROW OF  '=' SIGNS,  THE NEXT SIX LINES
C              CONTAIN METASTABLE LEVEL  INFORMATION, SUCH AS NUMBER OF
C              LEVELS, PARENT/SPIN INDEXES ETC. THE EIGHTH LINE CONTAINS
C              A ROW OF "-" SIGNS.  THIS DIFFERENCE IS USED TO IDENTIFY
C              WHICH FILE TYPE IS BEING READ.
C           B) THE PARTIAL FILES INCLUDE THE POPULATIONS  FOR  EACH  OF
C              THE METASTABLE LEVELS, WHEREAS THE STANDARD FILES CONTAIN
C              A SINGLE SET OF COMBINED POPULATIONS.
C
C          THE METASTABLE POPULATION FILES HAVE THE SAME FORMAT AS THE
C          PARTIAL MASTER CONDENSED FILES
C
C          THE CHARACTER STRING SEPARATING THE INPUT DATA FOR EACH
C          CHARGE STATE IN THE FILE WILL GIVE:
C
C          THE CHARGE STATE VALUE (Z1=) AND DATE (DATE:).
C                (OLDER DATA SETS MAY HAVE 'Z =' INSTEAD OF 'Z1=' HERE)
C          (CHARGE STATE Z1 = ION CHARGE + 1 = RECOMBINING ION CHARGE)
C
C          UNDER EACH OF THESE LINES THE COEFFTS/POPULATIONS ARE LISTED
C          -IN THE CASE OF THE METASTABLE/PARTIAL FILES VALUES FOR EACH
C          OF THE METASTABLE LEVELS ARE LISTED EACH BEING  PRECEDED  BY
C          A LINE GIVING THE METASTABLE INDEX BETWEEN TWO "/".
C          E.G.  /2/ => METASTABLE LEVEL TWO.
C
C          DATA FOR INDIVIDUAL PARENT/SPIN SYSTEMS ARE LISTED IN THE
C          MEATASTABLE FILES.
C
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  IUNIT   = UNIT TO WHICH INPUT DATA SET ALLOCATED
C  INPUT : (C*(*))DSNAME  = INPUT MASTER CONDENSED FILE DATA SET NAME
C  OUTPUT: (L*4)  LERROR  = .TRUE.  => ERROR DETECTED IN READING FILE
C                         = .FALSE. => NO ERROR DETECTED IN FILE
C
C  INPUT : (I*4)  NDDEN   = MAX. NUMBER OF REDUCED DENSITIES ALLOWED IN
C                           MASTER CONDENSED/METASTABLE FILE FOR A GIVEN
C                           SEQUENCE.
C  INPUT : (I*4)  NDTIN   = MAX. NO. OF REDUCED TEMPERATURES ALLOWED IN
C                           MASTER CONDENSED/METASTABLE FILE FOR A GIVEN
C                           SEQUENCE.
C  INPUT : (I*4)  NDZ1V   = MAX. NUMBER OF CHARGE STATES ALLOWED IN
C                           MASTER CONDENSED/METASTABLE FILE FOR A GIVEN
C                           SEQUENCE.
C  INPUT : (I*4)  NDMET   = MAX.  NUMBER OF METASTABLE STATES ALLOWED IN
C                           MASTER CONDENSED/METASTABLE FILE FOR A GIVEN
C                           SEQUENCE.
C                           NOT USED FOR STANDARD MASTER CONDENSED FILES
C                           (SET EQUAL TO 1 IN THIS CASE).
C
C  OUTPUT: (I*4)  IDE     = NUMBER OF REDUCED DENSITIES READ FROM INPUT
C                           MASTER CONDENSED/METASTABLE FILE FOR A GIVEN
C                           SEQUENCE.
C  OUTPUT: (I*4)  ITE     = NO. OF REDUCED TEMPERATURES READ FROM INPUT
C                           MASTER CONDENSED/METASTABLE FILE FOR A GIVEN
C                           SEQUENCE.
C  OUTPUT: (I*4)  IZE     = NO. OF CHARGE STATES GIVEN IN THE INPUT
C                           MASTER CONDENSED/METASTABLE FILE FOR A GIVEN
```

```
C                           SEQUENCE.
C
C  OUTPUT: (R*8)  DENSR() = SET OF 'IDE' INPUT REDUCED DENSITIES (CM-3/
C                           Z1**7) READ FROM CONDENSED MASTER/METASTABLE
C                           FILE.
C  OUTPUT: (R*8)  TR()    = SET OF 'ITE'  INPUT  REDUCED  TEMPERATURES
C                           (K/Z1**2) READ FROM CONDENSED MASTER/
C                           METASTABLE FILE.
C  OUTPUT: (R*8)  ZIPT()  = SET OF 'IZE' INPUT CHARGE STATES READ FROM
C                           CONDENSED MASTER/METASTABLE FILE.
C                           (CHARGE STATE = ION CHARGE + 1 = RECOMBINING
C                            ION CHARGE)
C
C  OUTPUT: (I*4)  IME     = NO. OF METASTABLE LEVELS CONTAINED IN THE
C                           INPUT MASTER CONDENSED/METASTABLE FILE.
C                           EQUALS 1 FOR STANDARD MASTER CONDENSED FILES
C  OUTPUT: (I*4)  IMETR() =THE ORIGINAL COPDAT INDEX FOR EACH METASTABLE
C                           LEVEL. DIMENSION: METASTABLE LEVEL INDEX.
C                           NOT USED FOR STANDARD MASTER CONDENSED FILES
C  OUTPUT: (C*12) CSTRGA()=THE DESIGNATION OF EACH METASTABLE LEVEL.
C                           DIMENSION: METASTABLE LEVEL INDEX.
C                           NOT USED FOR STANDARD MASTER CONDENSED FILES
C
C  OUTPUT: (I*4)  NPRNT   = NUMBER OF PARENTS CONTAINED IN THE INPUT
C                           MASTER CONDENSED/METASTABLE FILE.
C                           NOT USED FOR STANDARD MASTER CONDENSED FILES
C                           (NOTE: THE NUMBER OF PARENTS CANNOT EXCEED
C                                  THE NUMBER OF METASTABLE LEVELS)
C  OUTPUT: (I*4)  IPRNT() = THE PARENT INDEX FOR INPUT PARENT.
C                           DIMENSION: PARENT/(METASTABLE LEVEL) INDEX.
C                           NOT USED FOR STANDARD MASTER CONDENSED FILES
C  OUTPUT: (I*4)  IPSYS() = THE SPIN SYSTEM REFERENCE FOR EACH INPUT
C                           PARENT.
C                           DIMENSION: PARENT/(METASTABLE LEVEL) INDEX.
C                           NOT USED FOR STANDARD MASTER CONDENSED FILES
C
C  OUTPUT: (L*4)  LSWIT   = .TRUE.  => IONISATION POTENTIALS
C                                      INCLUDED IN INPUT MASTER FILE.
C                           .FALSE. => IONISATION POTENTIALS
C                                      NOT INCLUDED IN INPUT MASTER FILE
C  OUTPUT: (R*8)  EIA()   = IONISATION POTENTIALS: ()=ION CHARGE
C                           UNITS: WAVE NUMBERS (CM-1)
C                           (= 0.0 IF NOT SET)
C
C  OUTPUT: (R*8) AIPT(,,,)= OPTION 6: TOTAL    LINE POWER COEFFICIENTS
C                           OPTION 7: SPECIFIC LINE POWER COEFFICIENTS
C                           OPTION 8: METASTABLE STATE POPULATIONS
C                           1ST DIMENSION: ELECTRON DENSITY INDEX
C                                          ('DENSR()')
C                           2ND DIMENSION: ELECTRON TEMPERATURE INDEX
C                                          ('TR()')
C                           3RD DIMENSION: CHARGE STATE INDEX
C                                          ('ZIPT()')
C                           4TH DIMENSION: METASTABLE STATE INDEX
C                           (OPTIONS 6 & 7 STANDARD FILES ALWAYS = 1)
C
C          (C*1)  CBLNK   = PARAMETER = ' '
C          (C*1)  CEQUAL  = PARAMETER = '='
C          (C*1)  CSTAR   = PARAMETER = '*'
C
C          (I*4)  I4UNIT  = FUNCTION (SEE ROUTINE SECTION BELOW)
```

```
C          (I*4)  IPOT    = NUMBER OF IONISATION POTENTIAL VALUES
C                           PRESENT IN THE INPUT FILE.
C          (I*4)  IZ1     = CHARGE STATE READ FROM THE LINE PRECEEDING
C                           AN INPUT BLOCK FROM THE FILE.
C                           (= ION CHARGE + 1 = RECOMBINING ION CHARGE)
C          (I*4)  IMET    = METASTABLE STATE OF CURRENT DATA BLOCK BEING
C                           READ.
C          (I*4)  IBGN    = FIRST BYTE OF INTEREST IN CHARACTER 'STRING'
C          (I*4)  IEND    = LAST  BYTE OF INTEREST IN CHARACTER 'STRING'
C          (I*4)  ID      = ARRAY SUBSCRIPT USED FOR DENSITY INDEXES
C          (I*4)  IT      = ARRAY SUBSCRIPT USED FOR TEMPERATURE INDEXS
C          (I*4)  IZ      = ARRAY SUBSCRIPT USED FOR ION-CHARGE INDEXES
C          (I*4)  IM      = ARRAY SUBSCRIPT USED FOR METASTABLE INDEXES
C          (I*4)  I       = GENERAL USE
C
C          (L*4)  LPART   = .TRUE.  => REQUESTED INPUT FILE: PARTIAL
C                                                 OR METASTABLE POPULATION.
C                           = .FALSE. => REQUESTED INPUT FILE: STANDARD
C
C          (C*1)  C1      = GENERAL USE 1-BYTE CHARACTER STRING.
C                           (STORES METASTABLE STATE ORDER INDEX).
C          (C*5)  CPOT    = 'IPOT'
C
C          (C*80) STRING()= STRINGS  INTO WHICH LINES OF INPUT FILE ARE
C                           READ TO ENABLE ITS FORMAT TO BE ESTABLISHED
C                           AND CONTENTS READ.
C
C
C NOTE:
C          STREAM HANDLING:
C             STREAM 'IUNIT' IS USED FOR READING CONDENSED MASTER FILES
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          -------------------------------------------------------------
C          I4UNIT     ADAS      FETCH UNIT NUMBER FOR OUTPUT OF MESSAGES
C          XXREIA     ADAS      READ IN UNKNOWN NUMBER OF 'EIA' VALUES
C                               IF PRESENT.
C
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 4569
C
C DATE:    05/03/91 (DIFFERENT FROM ADAS90 VERSION - REMOVED DSN OPEN)
C
C UPDATE:  23/04/93 - PE BRIDEN - ADAS91: ADDED I4UNIT FUNCTION TO WRITE
C                                         STATEMENTS FOR SCREEN MESSAGES
C
C UPDATE:  24/05/93 - PE BRIDEN - ADAS91: CHANGED I4UNIT(0)-> I4UNIT(-1)
C
C UPDATE:  11/08/93 - HP SUMMERS - RENAMED TO XXIN80 FROM XXIN68
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 06-09-95
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST RELEASE (NO CHANGES)
C
C VERSION  : 1.2
C DATE     : 10-04-2007
```

```
C MODIFIED : Allan Whiteford
C               - Modified documentation as part of automated
C   subroutine documentation preparation.
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      CHARACTER*12        CSTRGA(NDMET)
      CHARACTER*(*)       DSNAME
      INTEGER             IDE,          IME,          IMETR(NDMET)
      INTEGER             IPRNT(NDMET),               IPSYS(NDMET)
      INTEGER             ITE,          IUNIT,        IZE,          NDDEN
      INTEGER             NDMET,        NDTIN,        NDZ1V,        NPRNT
      LOGICAL             LERROR,       LSWIT
      REAL*8              AIPT(NDDEN,NDTIN,NDZ1V,NDMET)
      REAL*8              DENSR(NDDEN),               EIA(250)
      REAL*8              TR(NDTIN),    ZIPT(NDZ1V)
```

9.143 xxindx: Subroutine xxindx from library adaslib

```
      SUBROUTINE XXINDX( N , ARR , INDX )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXINDX *********************
C
C  PURPOSE: INDEXES AN ARRAY 'ARR' OF LENGTH 'N'. IT OUTPUTS THE ARRAY
C           'INDX()' SUCH THAT 'ARR(INDX(J))' IS  IN  ASCENDING  ORDER
C           FOR J=1,2,...,N.   THE INPUT QUANTITIES 'N' AND 'ARR'  ARE
C           NOT CHANGED.
C
C  REFERENCE:  NUMERICAL RECIPES: The Art of Scientific Computing
C              (FORTRAN Version).
C              W.H.Press, B.P.Flannery, S.A.Teukolsky & W.T.Vetterling.
C              (Cambridge University Press, Cambridge). 1989. p.233
C              ISBN 0 521 38330 7
C
C  CALLING PROGRAM: GENERAL USE
C
C  SUBROUTINE:
C
C  INPUT :    (I*4)  N       = INPUT ARRAY 'ARR()' LENGTH
C  INPUT :    (R*8)  ARR()   = ARRAY TO BE INDEXED
C  OUTPUT:    (I*4)  INDX()  = ASCENDING ORDER INDEX
C
C             (R*8)  INDXT   = TEMPORARY STORAGE FOR 'INDX' ELEMENT
C             (I*4)  L       = "HIRING" PHASE INDEX
C             (I*4)  IR      = "RETIREMENT-AND-PROMOTION"  PHASE INDEX
C             (I*4)  I       = GENERAL USE
C             (I*4)  J       = GENERAL USE
C
C             (R*8)  Q       = TEMPORARY STORAGE FOR 'ARR' ELEMENT
C
C ROUTINES:  NONE
C
C NOTE:      USES THE HEAPSORT METHOD.
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 4569
C
C DATE:    09/10/90
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             INDX(N),     N
      REAL*8              ARR(N)
```

9.144 xxinst: Subroutine xxinst from library adaslib

```
      SUBROUTINE XXINST( IUNIT  , DSNAME , LERROR ,
     &                   NDDEN  , NDTIN  , NDZ1V  ,
     &                   IDE    , ITE    , IZE    ,
     &                   DENSR  , TR     , ZIPT   ,
     &                   LSWIT  , EIA    ,
     &                   AIPT
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXINST *********************
C
C  PURPOSE: TO FETCH DATA FROM STANDARD MASTER CONDENSED
C           COLLISIONAL-DIELECTRONIC FILES.
C
C           THE FOLLOWING FILES ARE ALLOWED:
C
C             1. RECOMBINATION COEFFICIENTS
C             2. IONISATION    COEFFICIENTS
C             3. CHARGE-EXCHANGE RECOMBINATION COEFFICIENTS
C             4. RECOMBINATION-BREMSSTRAHLUNG  POWER COEFFICIENTS
C             5. CHARGE-EXCHANGE RECOMBINATION POWER COEFFICIENTS
C             6. TOTAL    LINE POWER COEFFICIENTS
C             7. SPECIFIC LINE POWER COEFFICIENTS
C
C             (NOTE: PARTIAL MASTER CONDENSED FILES SHOULD BE READ USING
C                    'XXIN15' & 'XXIN68'.
C                    IF  PARTIAL AND STANDARD FILES ARE BOTH TO BE READ
C                    USE 'XXIN15' AND 'XXIN68')
C
C  CALLING PROGRAM: GENERAL USE
C
C  DATA:
C           THE  SOURCE  DATA IS CONTAINED AS  MEMBERS  OF  PARTITIONED
C           DATA SETS AS FOLLOWS:
C
C             1. JETUID.ACD<YR>.DATA
C             2. JETUID.SCD<YR>.DATA
C             3. JETUID.CCD<YR>.DATA
C             4. JETUID.PRB<YR>.DATA
C             5. JETUID.PRC<YR>.DATA
C             5. JETUID.PLT<YR>.DATA
C             5. JETUID.PLS<YR>.DATA
C
C           WHERE <YR> DENOTES TWO INTEGERS FOR THE YEAR SELECTED.
C           IF <YR> IS BLANK THEN THE CURRENT RECOMMENDED DATA SETS ARE
C           USED
C
C           THE MEMBERS OF THE PARTITIONED DATA SETS MUST BE STANDARD
C           MASTER CONDENSED FILES AND OF THE FORM <SE>
C
C           WHERE: <SE> IS THE ONE OR TWO LETTER ION SEQUENCE CODE.
C
C                  E.G.  'C'   OR 'HE'  ('@' = BARE-NUCLEUS)
C
C                  PARTIAL  FILES WHICH HAVE MEMBERS NAMES WITH EITHER
C                  TWO DIGITS OR A DOUBLE HASH  AFTER THEM  CANNOT  BE
C                  READ USING THIS ROUTINE (SEE 'XXIN15' & 'XXIN68')
C
C           THE  'PARTIAL'  AND  'STANDARD'  MASTER CONDENSED FILES  ARE
```

```
C           SIMILAR IN FORM. A MAJOR DIFFERENCE IS THAT ADDITIONAL LINES
C           ARE  INCLUDED  AT  THE  BEGINNING OF  THE  'PARTIAL'  MASTER
C           FILES.    THE  FIRST OF THESE LINES CONTAINING A ROW OF  '='
C           SIGNS.  THIS DIFFERENCE IS USED TO MAKE SURE THAT A STANDARD
C           MASTER FILE IS BEING READ.
C
C           THE CHARACTER STRING SEPARATING THE INPUT DATA FOR EACH
C           VALUE OF Z1 IN THE FILE WILL GIVE:
C
C                        THE Z1 VALUE (Z1=) AND DATE (DATE:).
C              (OLDER DATA SETS MAY HAVE 'Z =' INSTEAD OF 'Z1=' HERE)
C
C           (NOTE: Z1 = ION CHARGE + 1 = RECOMBINING ION CHARGE)
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  IUNIT   = UNIT TO WHICH INPUT DATA SET ALLOCATED
C  INPUT : (C*(*))DSNAME  = INPUT MASTER CONDENSED FILE DATA SET NAME
C  OUTPUT: (L*4)  LERROR  = .TRUE.  => ERROR DETECTED IN READING FILE
C                         = .FALSE. => NO ERROR DETECTED IN FILE
C
C  INPUT : (I*4)  NDDEN   = MAX. NUMBER OF REDUCED DENSITIES ALLOWED IN
C                           MASTER CONDENSED FILE FOR A GIVEN SEQUENCE
C  INPUT : (I*4)  NDTIN   = MAX. NO. OF REDUCED TEMPERATURES ALLOWED IN
C                           MASTER CONDENSED FILE FOR A GIVEN SEQUENCE
C  INPUT : (I*4)  NDZ1V   = MAX. NUMBER OF CHARGE STATES ALLOWED IN
C                           MASTER CONDENSED FILE FOR A GIVEN SEQUENCE
C
C  OUTPUT: (I*4)  IDE     = NUMBER OF REDUCED DENSITIES READ FROM INPUT
C                           MASTER CONDENSED FOR A GIVEN SEQUENCE
C  OUTPUT: (I*4)  ITE     = NO. OF REDUCED TEMPERATURES READ FROM INPUT
C                           MASTER CONDENSED FOR A GIVEN SEQUENCE
C  OUTPUT: (I*4)  IZE     = NO. OF CHARGE STATES GIVEN IN THE INPUT
C                           MASTER CONDENSED FOR A GIVEN SEQUENCE
C
C  OUTPUT: (R*8)  DENSR() = SET OF 'IDE' INPUT REDUCED DENSITIES (CM-3/
C                           Z1**7) READ FROM CONDENSED MASTER FILE.
C  OUTPUT: (R*8)  TR()    = SET OF 'ITE'  INPUT  REDUCED  TEMPERATURES
C                           (K/Z1**7) READ FROM CONDENSED MASTER FILE.
C  OUTPUT: (R*8)  ZIPT()  = SET OF 'IZE' INPUT CHARGE STATES READ FROM
C                           CONDENSED MASTER/METASTABLE FILE.
C                           (CHARGE STATE = ION CHARGE + 1 = RECOMBINING
C
C  OUTPUT: (L*4)  LSWIT   = .TRUE.  => IONISATION RATE COEFFICIENTS
C                                      INCLUDED IN INPUT MASTER FILE.
C                           .FALSE. => IONISATION RATE COEFFICIENTS
C                                      NOT INCLUDED IN INPUT MASTER FILE
C  OUTPUT: (R*8)  EIA()   = IONISATION RATE COEFFICIENTS: ()=ION CHARGE
C                           UNITS: WAVE NUMBERS (CM-1)
C                           (= 0.0 IF NOT SET )
C
C  OUTPUT: (R*8)  AIPT(,,)= CONDENSED MASTER FILE DATA. COLL-DIEL COEFF.
C                           1ST DIMENSION: REDUCED DENSITY ('DENSR()')
C                           2ND DIMENSION: REDUCED TEMPERATURE ('TR()')
C                           3RD DIMENSION: CHARGE STATE ('ZIPT()')
C
C          (I*4)  I4UNIT  = FUNCTION (SEE ROUTINE SECTION BELOW)
C          (I*4)  IPOT    = NUMBER OF IONISATION RATE COEFF. VALUES
C                           PRESENT IN THE INPUT FILE.
C          (I*4)  IZ1     = CHARGE STATE READ FROM THE LINE PRECEEDING
C                           AN INPUT BLOCK FROM THE FILE.
```

```
C                             (= ION CHARGE + 1 = RECOMBINING ION CHARGE)
C          (I*4)  IBGN    = FIRST BYTE OF INTEREST IN CHARACTER 'STRING'
C          (I*4)  IEND    = LAST  BYTE OF INTEREST IN CHARACTER 'STRING'
C          (I*4)  ID      = ARRAY SUBSCRIPT USED FOR DENSITY VALUES
C          (I*4)  IT      = ARRAY SUBSCRIPT USED FOR TEMPERATURE VALUES
C          (I*4)  IZ      = ARRAY SUBSCRIPT USED FOR ION-CHARGE VALUES
C          (I*4)  I       = GENERAL USE
C
C          (C*5)  CPOT    = 'IPOT'
C          (C*80) STRING  = STRING INTO WHICH 1ST LINE OF INPUT FILE IS
C                           READ TO ENABLE ITS FORMAT TO BE ESTABLISHED.
C
C NOTE:
C          STREAM HANDLING:
C             STREAM 'IUNIT' IS USED FOR READING CONDENSED MASTER FILES
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4UNIT     ADAS      FETCH UNIT NUMBER FOR OUTPUT OF MESSAGES
C          XXREIA     ADAS      READ IN UNKNOWN NUMBER OF 'EIA' VALUES
C                               IF PRESENT.
C
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 4569
C
C DATE:    22/08/90
C
C DATE:    05/03/90 - PE BRIDEN - ADAS91: OPENING OF DATA SET REMOVED.
C
C UPDATE:  23/04/93 - PE BRIDEN - ADAS91: ADDED I4UNIT FUNCTION TO WRITE
C                                         STATEMENTS FOR SCREEN MESSAGES
C
C UPDATE:  24/05/93 - PE BRIDEN - ADAS91: CHANGED I4UNIT(0)-> I4UNIT(-1)
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 06-09-95
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST RELEASE (NO CHANGES)
C
C VERSION  : 1.2
C DATE     : 10-04-2007
C MODIFIED : Allan Whiteford
C               - Modified documentation as part of automated
C   subroutine documentation preparation.
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      CHARACTER*(*)       DSNAME
      INTEGER             IDE,         ITE,         IUNIT,       IZE
      INTEGER             NDDEN,       NDTIN,       NDZ1V
      LOGICAL             LERROR,      LSWIT
      REAL*8              AIPT(NDDEN,NDTIN,NDZ1V),  DENSR(NDDEN)
      REAL*8              EIA(50),     TR(NDTIN),   ZIPT(NDZ1V)
```

9.145 xxisrt: Subroutine xxisrt from library adaslib

```
      SUBROUTINE XXISRT( N , ARR , INDX , WRK )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXISRT *********************
C
C  PURPOSE: SORTS AN ARRAY 'ARR' OF LENGTH 'N' ACCORDING TO THE INDEX
C           'INDX()'  SUCH  THAT  'ARR(INDX(J))'  GOES  TO   'ARR(J)'
C           FOR J=1,2,...,N.   THE INPUT QUANTITIES 'N' AND 'INDX' ARE
C           NOT CHANGED.
C
C  CALLING PROGRAM: GENERAL USE
C
C  SUBROUTINE:
C
C  INPUT :    (I*4)  N       = INPUT ARRAY 'ARR()' LENGTH
C  I/O   :    (R*8)  ARR()   = ARRAY TO BE SORTED
C  INPUT :    (I*4)  INDX()  = RE-ORDERING INDEX
C  OUTPUT:    (R*8)  WRK()   = WORKSPACE
C
C             (I*4)  I       = GENERAL USE
C
C ROUTINES:  NONE
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 4569
C
C DATE:    11/07/90
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             INDX(N),     N
      REAL*8              ARR(N),      WRK(N)
```

9.146 xxlast: Subroutine xxlast from library adaslib

```
      subroutine xxlast(cstrg, csrch, indx)

c-----------------------------------------------------------------------
c
c  ***************** fortran77 function: xxlast **********************
c
c  purpose : returns the index of the last occurence of a character in
c            a string
c
c  calling program: general use
c
c  subroutine:
c
c  input : (c*(*)) cstrg    = input string for interrogation
c  input : (c*1)   csrch    = search character
c
c  output: (i*4)   indx     = byte position of last  occurence
c                             of csrch in input string.
c
c          (i*4)   i        = general use
c          (i*4)   ilen     = length of 'cstrng' string in bytes
c
c routines:
c          routine    source    brief description
c          ------------------------------------------------------------
c          i4unit     adas      fetch unit number for output of messages
c
c author:  Hugh Summers
c          JET K1/1/61
c          Tel. 01235-46-4459
c date:    19-12-2007
c
c
c version  : 1.1
c date     : 19-12-2007
c modified : Hugh Summers
c              - first version.
c-----------------------------------------------------------------------
      CHARACTER           CSRCH
      CHARACTER*(*)       CSTRG
      INTEGER             INDX
```

9.147 xxlim4: Subroutine xxlim4 from library adaslib

```
       SUBROUTINE XXLIM4 ( GMIN , GMAX , X , N , CUTMIN )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXLIM4 *********************
C
C  PURPOSE:
C             FINDS MAXIMA AND MINIMA OF A LIST  OF X VALUES AND RETURNS
C             THE SCALE RANGE FOR PLOTTING ON A LOG TO THE BASE 10 GRID.
C             I.E. LOG10 RANGE TO NEAREST APPROPRIATE INTEGERS.
C
C             (RANGE IN LINEAR UNITS = GMIN -> GMAX)
C             (REAL VERSION OF 'XXLIM8')
C
C  CALLING PROGRAM: GENERAL USE
C
C  SUBROUTINE:
C
C  OUTPUT: (R*4)  GMIN    = LOWER LIMIT FOR GRAPH X-AXIS (LINEAR)
C  OUTPUT: (R*4)  GMAX    = UPPER LIMIT FOR GRAPH X-AXIS (LINEAR)
C  INPUT : (R*4)  X()     = INPUT X-VALUES
C  INPUT : (I*4)  N       = NUMBER OF INPUT X-VALUES
C  INPUT : (R*4)  CUTMIN  = MINIMUM ALLOWED VALUE FOR 'GMIN' (LINEAR).
C                           ('CUTMIN=0' EFFECTIVELY REMOVES ITS EFFECT)
C
C          (I*4)  NCMAX   =  PARAMETER = MAXIMUM NUMBER OF LOG10 CYCLES
C                                        ALLOWED IN GHOST80
C          (I*4)  IMIN    = MINIMUM LOG10 CYCLE
C          (I*4)  IMAX    = MAXIMUM LOG10 CYCLE
C          (I*4)  I       = GENERAL USE
C
C AUTHOR:    PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C            K1/0/81
C            JET EXT. 4569
C
C DATE:      02/08/90
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 06-09-95
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST RELEASE (NO CHANGES)
C VERSION  : 1.2
C DATE     : 20-12-2001
C MODIFIED : Martin O'Mullane
C               - Removed mainframe listing information beyond column 72.
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      INTEGER             N
      REAL                CUTMIN,      GMAX,        GMIN,        X(N)
```

9.148 xxlim8: Subroutine xxlim8 from library adaslib

```
       SUBROUTINE XXLIM8 ( GMIN , GMAX , X , N , CUTMIN )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXLIM8 *********************
C
C  PURPOSE:
C             FINDS MAXIMA AND MINIMA OF A LIST  OF X VALUES AND RETURNS
C             THE SCALE RANGE FOR PLOTTING ON A LOG TO THE BASE 10 GRID.
C             I.E. LOG10 RANGE TO NEAREST APPROPRIATE INTEGERS.
C
C             (RANGE IN LINEAR UNITS = GMIN -> GMAX)
C             (DOUBLE PRECISION VERSION OF 'XXLIM4')
C
C  CALLING PROGRAM: GENERAL USE
C
C  SUBROUTINE:
C
C  OUTPUT: (R*4)  GMIN    = LOWER LIMIT FOR GRAPH X-AXIS (LINEAR)
C  OUTPUT: (R*4)  GMAX    = UPPER LIMIT FOR GRAPH X-AXIS (LINEAR)
C  INPUT : (R*8)  X()     = INPUT X-VALUES
C  INPUT : (I*4)  N       = NUMBER OF INPUT X-VALUES
C  INPUT : (R*4)  CUTMIN  = MINIMUM ALLOWED VALUE FOR 'GMIN' (LINEAR).
C                           ('CUTMIN=0' EFFECTIVELY REMOVES ITS EFFECT)
C
C          (I*4)  NCMAX   =  PARAMETER = MAXIMUM NUMBER OF LOG10 CYCLES
C                                        ALLOWED IN GHOST80
C
C          (I*4)  IMIN    = MINIMUM LOG10 CYCLE
C          (I*4)  IMAX    = MAXIMUM LOG10 CYCLE
C          (I*4)  I       = GENERAL USE
C
C AUTHOR:    PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C            K1/0/81
C            JET EXT. 4569
C
C DATE:      02/08/90
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 06-09-95
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST RELEASE (NO CHANGES)
C
C VERSION  : 1.2
C DATE     : 20-12-2001
C MODIFIED : Martin O'Mullane
C               - Removed mainframe listing information beyond column 72.
C
C VERSION  : 1.3
C DATE     : 10-04-2007
C MODIFIED : Allan Whiteford
C               - Modified documentation as part of automated
C   subroutine documentation preparation.
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      INTEGER             N
      REAL                CUTMIN,      GMAX,        GMIN
      REAL*8              X(N)
```

9.149 xxlm28: Subroutine xxlm28 from library adaslib

```
       SUBROUTINE XXLM28 ( GMIN   , GMAX   ,
     &                     Y      ,
     &                     NDIM1  , NDIM2  ,
     &                     LMAX1  , LMAX2  ,
     &                     CUTMIN
     &                   )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXLM28 *********************
C
C  PURPOSE:
C           FINDS MAXIMA AND MINIMA OF A 2-DIMENSIONAL ARRAY OF Y-VALUES
C           AND  RETURNS THE SCALE RANGE FOR PLOTTING  ON A LOG  TO  THE
C           BASE 10 GRID.
C
C           I.E. LOG10 RANGE COVERING BOTH DIMENSIONS TO THE NEAREST
C                APPROPRIATE INTEGERS.
C           (RANGE IN LINEAR UNITS = GMIN -> GMAX)
C           (REAL*8 VERSION OF 'XXLM24')
C
C  CALLING PROGRAM: GENERAL USE
C
C  SUBROUTINE:
C
C  OUTPUT: (R*4)  GMIN    = LOWER LIMIT FOR GRAPH Y-AXIS (LINEAR)
C  OUTPUT: (R*4)  GMAX    = UPPER LIMIT FOR GRAPH Y-AXIS (LINEAR)
C
C  INPUT : (R*8)  Y(,)    = TWO-DIMENSIONAL ARRAY OF INPUT Y-VALUES.
C
C  INPUT : (I*4)  NDIM1   = FIRST 'Y(,)' ARRAY DIMENSION BOUND
C  INPUT : (I*4)  NDIM2   = SECOND 'Y(,)' ARRAY DIMENSION BOUND
C
C  INPUT : (I*4)  LMAX1   = RANGE OF FIRST 'Y(,)' ARRAY DIMENSION
C                           TO BE ASSESSED (1 -> LMAX1) (LMAX1<=NDIM1)
C  INPUT : (I*4)  LMAX2   = RANGE OF SECOND 'Y(,)' ARRAY DIMENSION
C                           TO BE ASSESSED (1 -> LMAX2) (LMAX2<=NDIM2)
C
C  INPUT : (R*4)  CUTMIN  = MINMUM VALUE FOR 'GMIN' (LINEAR).
C                           ('CUTMIN=0' EFFECTIVELY REMOVES ITS EFFECT)
C
C          (I*4)  NCMAX   = PARAMETER = MAXIMUM ALLOWED NUMBER OF LOG10
C                                       CYCLES ALLOWED IN GHOST80
C
C          (I*4)  IMIN    = MINIMUM LOG10 CYCLE
C          (I*4)  IMAX    = MAXIMUM LOG10 CYCLE
C          (I*4)  I       = GENERAL USE
C          (I*4)  J       = GENERAL USE
C
C
C NOTES:
C        A MINIMUM AVALUE FOR 'GMIN' IS SPECIFIED IN THE ARGUMENTS.
C        (SEE 'CUTMIN').
C
C ROUTINES: NONE
C
C AUTHOR:   PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C           K1/0/81
C           JET EXT. 4569
C
```

```
C DATE:     02/08/90
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 06-09-95
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST RELEASE(NO CHANGES MADE)
C
C VERSION  : 1.2
C DATE     : 10-04-2007
C MODIFIED : Allan Whiteford
C               - Modified documentation as part of automated
C   subroutine documentation preparation.
C-----------------------------------------------------------------------
      INTEGER             LMAX1,       LMAX2,       NDIM1,       NDIM2
      REAL                CUTMIN,      GMAX,        GMIN
      REAL*8              Y(NDIM1,NDIM2)
```

9.150 xxmadd: Subroutine xxmadd from library adaslib

```
      SUBROUTINE XXMADD( NR , NC , X , A , Y , B , C )
C
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXVADD *********************
C
C  PURPOSE: ADDS TWO MATRICES WITH MULTIPLIER FOR EACH.
C
C  CALLING PROGRAM: GENERAL USE.
C
C  INPUT : (I*4)  NR      = NUMBER OF ROWS IN MATRICES 'A' AND 'B'.
C  INPUT : (I*4)  NC      = NUMBER OF COLUMNS IN MATRICES 'A' AND 'B'.
C  INPUT : (R*8)  X       = FIRST MULTIPLIER.
C  INPUT : (R*8)  A(,)    = FIRST MATRIX.
C                           1ST DIMENSION: NR
C                           2ND DIMENSION: NC
C  INPUT : (R*8)  Y       = SECOND MULTIPLIER.
C  INPUT : (R*8)  B(,)    = SECOND MATRIX.
C                           1ST DIMENSION: NR
C                           2ND DIMENSION: NC
C
C  OUTPUT: (R*8)  C       = OUTPUT MATIRX.
C                           1ST DIMENSION: NR
C                           2ND DIMENSION: NC
C
C          (I*4)  I       = LOOP INDEX.
C          (I*4)  J       = LOOP INDEX.
C
C ROUTINES: NONE
C
C AUTHOR:  JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 5183
C
C DATE:    10/11/93
C
C UNIX-IDL PORT:
C
C AUTHOR:  WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C
C DATE:    22ND MAY 1996
C
C VERSION: 1.1                          DATE: 22-05-96
C MODIFIED: WILLIAM OSBORN
C               - FIRST VERSION. IBM VERSION NOT CHANGED
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             NC,          NR
      REAL*8              A(NR,NC),    B(NR,NC),    C(NR,NC),    X
      REAL*8              Y
```

9.151 xxmcpy: Subroutine xxmcpy from library adaslib

```
      SUBROUTINE XXMCPY( NR , NC , A , B )
C
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXVADD *********************
C
C  PURPOSE: COPIES ONE MATRIX TO ANOTHER.
C
C  CALLING PROGRAM: GENERAL USE.
C
C  INPUT : (I*4)  NR      = NUMBER OF ROWS IN MATRICES 'A' AND 'B'.
C  INPUT : (I*4)  NC      = NUMBER OF COLUMNS IN MATRICES 'A' AND 'B'.
C  INPUT : (R*8)  A(,)    = INPUT MATRIX.
C                           1ST DIMENSION: NR
C                           2ND DIMENSION: NC
C
C  OUTPUT: (R*8)  B(,)    = OUTPUT MATRIX.
C                           1ST DIMENSION: NR
C                           2ND DIMENSION: NC
C
C          (I*4)  I       = LOOP INDEX.
C          (I*4)  J       = LOOP INDEX.
C
C ROUTINES: NONE
C
C AUTHOR:  JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 5183
C
C DATE:    11/11/93
C
C UNIX-IDL PORT:
C
C AUTHOR:  WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C
C DATE:    22ND MAY 1996
C
C VERSION: 1.1                          DATE: 22-05-96
C MODIFIED: WILLIAM OSBORN
C               - FIRST VERSION. IBM VERSION NOT CHANGED
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             NC,          NR
      REAL*8              A(NR,NC),    B(NR,NC)
```

9.152 xxmerg: Subroutine xxmerg from library adaslib

```
       subroutine xxmerg( g1   , n1  ,
     &                    g2   , n2   ,
     &                    grid , num
     &                  )

C-----------------------------------------------------------------------
C
C ROUTINE: xxmerg
C
C PURPOSE: Merges two grids and eliminates any duplicate entries.
C
C CALLING PROGRAM: general use
C
C  input : (i*4)  ndeng    = maximum number of energies in adf35 file.
C  input : (i*4)  ndedge   = maximum number of energy edges in adf35 file.
C  input : (i*4)  ieng     = actual number of energies.
C
C
C  ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          d1merg     NETLIB    Merge two strings of ascending numbers.
C          dcopy      NETLIB    Copies a vector, x, to a vector, y.
C
C  NOTES: The netlib routines form part of this file.
C
C  VERSION  : 1.1
C  DATE     : 01-08-2003
C  MODIFIED : Martin O'Mullane
C              - First version.
C
C  VERSION  : 1.2
C  DATE     : 16-02-2005
C  MODIFIED : Martin O'Mullane
C              - Increase number of steps, initialize grid and perform
C                more checks when filling grid().
C
C-----------------------------------------------------------------------

C-----------------------------------------------------------------------
      INTEGER             N1,          N2,          NUM
      REAL*8              G1(*),       G2(*),       GRID(*)
      DOUBLE PRECISION    TCOS(*)
      INTEGER             I1,          I2,          I3,          M1
      INTEGER             M2
      INTEGER             INCX,        INCY,        N
      REAL*8              DX(*),       DY(*)
```

9.153 xxmino: Subroutine xxmino from library adaslib

```
      SUBROUTINE XXMINO( ND , A , B )
C
C
C----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXMINO *********************
C
C  PURPOSE: INVERTS A MATRIX.
C
C  CALLING PROGRAM: GENERAL USE.
C
C  INPUT : (I*4)  ND      = DIMENSION OF MATRICES 'A' AND 'B'.
C  INPUT : (R*8)  A(,)    = MATRIX TO BE INVERTED.
C                           1ST DIMENSION: ND
C                           2ND DIMENSION: ND
C
C  OUTPUT: (R*8)  B(,)    = INVERTED MATRIX.
C                           1ST DIMENSION: ND
C                           2ND DIMENSION: ND
C
C
C          (I*4)  I       = LOOP INDEX.
C          (I*4)  J       = LOOP INDEX.
C
C          (R*8)  DINT    = DUMMY ARGUMENT TO 'XXMINV'.
C
C          (L*4)  LSOLVE  = .FALSE. => 'XXMINV' ONLY INVERTS MATRIX.
C
C  OUTPUT: (R*8)  DUM()   = DUMMY ARGUMENT TO 'XXMINV'.
C                           DIMENSION: ND
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          XXMINV     ADAS      BOTH INVERTS A MATRIX AND SOLVES A SET
C                               OF SIMULTANEOUS LINEAR EQUATIONS.
C
C AUTHOR:  JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 5183
C
C DATE:    10/11/93
C
C UNIX-IDL PORT:
C
C AUTHOR:  WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C
C DATE:    22ND MAY 1996
C
C VERSION: 1.1                          DATE: 22-05-96
C MODIFIED: WILLIAM OSBORN
C               - FIRST VERSION. IBM VERSION NOT CHANGED
C
C----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             ND
      REAL*8              A(ND,ND),    B(ND,ND)
```

9.154 xxminv: Subroutine xxminv from library adaslib

```
      SUBROUTINE XXMINV( LSOLVE , NDMAX , NDIM   ,
     &                   A      , B     , DINT
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXMINV *********************
C
C  PURPOSE: MATRIX INVERSION WITH ACCOMPANYING SOLUTION OF LINEAR
C           EQUATIONS IF REQUESTED.
C
C
C  CALLING PROGRAM: GENERAL USE
C
C
C  SUBROUTINE:
C
C  INPUT : (L*4)  LSOLVE   = .TRUE.  => SOLVES  SET  OF   'N'   LINEAR
C                                       EQUATIONS A.X = B, WHERE A,X,B
C                                       ARE MATRICES/VECTORS AND:
C                                       A = 'A(,)' ON INPUT
C                                       X = 'B()'  ON OUTPUT
C                                       B = 'B()'  ON INPUT
C                            .FALSE. => DOES NOT SOLVE ANY EQUATIONS
C                                       AND 'B()' IS NOT USED.
C
C  INPUT : (I*4)  NDMAX    = MAXIMUM VALUE OF 'N' ALLOWED. I.E.
C                             PHYSICAL DIMENSION OF INPUT MATRICES.
C  INPUT : (I*4)  NDIM     = DIMENSIONALITY 'N' OF MATRIX/EQUATIONS
C                             TO BE SOLVED.
C
C  I/O   : (R*8)  A(,)     = 'N' BY 'N' MATRIX
C                             INPUT : MATRIX TO BE INVERTED
C                             OUTPUT: INVERTED MATRIX
C  I/O   : (R*8)  B()      = USED ONLY IF 'LSOLVE=.TRUE.'WHEN THE SET
C                             OF 'N' LINEAR EQUATIONS ARE TO BE SOLVED.
C                             A.X = B
C                             INPUT : RIGHT HAND SIDE VECTOR 'B'
C                             OUTPUT: SOLUTION VECTOR 'X'
C
C  OUTPUT: (R*8)  DINT     = +1 OR -1 DEPENDING ON WHETHER THE  NUMBER
C                             OF  ROW  INTERCHANGES  WAS  EVEN  OR  ODD,
C                             RESPECTIVELY.
C
C          (I*4)  NLMAX    = PARAMETER: MUST BE >= 'NDMAX'
C
C          (I*4)  LROW     = ROW OF INPUT 'A(*,)' CONTAINING LARGEST
C                             NON-ZERO ELEMENT.
C          (I*4)  LCOLUM   = COLUMN OF INPUT 'A(,*)' CONTAINING LARGEST
C                             NON-ZERO ELEMENT.
C                             'LROW' & 'LCOLUM' => PIVOT POSITION IN A(,)
C          (I*4)  I1       = ARRAY INDICES: 1ST LEVEL
C          (I*4)  I2       = ARRAY INDICES: 2ND LEVEL
C          (I*4)  I3       = ARRAY INDICES: 3RD LEVEL
C
C          (R*8)  DMAX     = LARGEST NON-ZERO ELEMENT OF 'A(,)',
C                             AND EQUALS 'ABS(A(LROW,LCOLUM))'
C          (R*8)  SWAP     = TEMPORARAY STORAGE WHEN INTERCHANGING
C                             ROWS/COLUMNS.
C          (R*8)  TVAL     = TEMPORARY STORAGE OF VALUES IN NON-PIVOT
```

```
C                             ROWS.
C
C          (I*4)  IPIVOT() = NUMBER OF TIMES PIVOT FOUND IN COLUMN
C                             DIMENSION: COLUMN INDEX
C          (I*4)  INDX(,1) = ROW OF PIVOT IN 'A(,)'
C                             DIMENSION: PIVOT INDEX
C          (I*4)  INDX(,2) = COLUMN OF PIVOT IN 'A(,)'
C                             DIMENSION: PIVOT INDEX
C
C          (R*8)  PIVOT()  = 'DMAX' FOR PIVOT: DIMENSION PIVOT INDEX
C
C
C ROUTINES: NONE
C
C NOTE: THIS IS A STRUCTURED VERSION OF THE SUBROUTINE 'MATINV' WRITTEN
C       BY HUGH SUMMERS.
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 4569
C
C DATE:    31/08/90
C
C UPDATE:  15/05/92 - PE BRIDEN - INCREASED NLMAX FROM 50 -> 75
C
C UPDATE:  26/06/92 - PE BRIDEN - INCREASED NLMAX FROM 75 -> 200
C
C VERSION: 1.2 DATE: 04-05-2000
C MODIFIED: RICHARD MARTIN
C CHANGED DOCUMENTATION FOR NDIM.
C
C VERSION  : 1.3
C DATE     : 20-12-2001
C MODIFIED : Martin O'Mullane
C               - Removed mainframe listing information beyond column 72.
C
C VERSION: 1.4                                  DATE: 02-03-2003
C MODIFIED: Martin O'Mullane & Allan Whiteford
C - Increased NLMAX to 801 and improved error message.
C
C VERSION: 1.5                                  DATE: 29-05-2003
C MODIFIED: Martin O'Mullane
C - Increased NLMAX to 1201.
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      INTEGER             NDIM,        NDMAX
      LOGICAL             LSOLVE
      REAL*8              A(NDMAX,NDMAX),           B(NDMAX),    DINT
```

9.155 xxmkrc: Subroutine xxmkrc from library adaslib

```
      subroutine  xxmkrc( ndcnct   ,
     &                    iz0      , iptnl    ,
     &                    ncnct    , icnctv
     &                  )
c-----------------------------------------------------------------------
c
c  ****************** fortran77 subroutine: xxmkrc *********************
c
c  purpose:  to create the root connection vector for an element
c
c  calling program: various
c
c
c  subroutine:
c
c  input : (i*4)  ndcnct      = maximum number of elements in
c                               connection vector
c  input : (i*4)  iz0         = nuclear charge
c  input : (i*4)  iptnl       = root partition level (0 or 1)
c
c
c  output: (i*4)  ncnct       = number of elements in connection vector
c  output: (i*4)  cnctv()     = values of elements of  connection vector
c
c  routines:
c          routine    source    brief description
c          ------------------------------------------------------------
c          i4unit     adas      fetch unit number for output of messages
c
c  author:  h. p. summers, university of strathclyde
c           ja7.08
c           tel. 0141-548-4196
c
c  date:    26/05/06
c
c version: 1.1                          date: 26/05/2006
c modified: Hugh P Summers
c               - first edition.
c
c version: 1.2                          date: 17/01/2007
c modified: Hugh P Summers
c               - corrected metastable count for Ne+0.
c
c-----------------------------------------------------------------------
c-----------------------------------------------------------------------
      INTEGER             ICNCTV(NDCNCT),           IPTNL,       IZ0
      INTEGER             NCNCT,       NDCNCT
```

9.156 xxmkrp: Subroutine xxmkrp from library adaslib

```
       subroutine xxmkrp( ndstack       ,
     &                    iz0           , iptnl       ,
     &                    ncptn_stack   , cptn_stack
     &                  )
c-----------------------------------------------------------------------
c
c ****************** fortran 77 subroutine: xxmkrp *********************
c
c purpose: To create a root partition and return the partition block.
c
c
c calling program: adas416
c
c subroutine:
c
c input : (i*4)   ndstack     = maximum partition block lines
c input : (i*4)   iz0         = nuclear charge
c input : (i*4)   iptnl       = root partition level (0 or 1)
c
c output: (i*4)  ncptn_stack  = number of lines in the partition block
c output: (c*80) cptn_stack() = character string lines of the partition
c                               block
c
c
c routines:
c    routine    source brief description
c    ------------------------------------------------------------
c   i4unit     adas      fetch unit number for output of messages
c         xxopen     adas      inquire, open and allocate file to unit
c         xxslen     adas      find non-blank characters in string
c         xxword     adas      extract position of number in buffer
c
c
c author:  h. p. summers, university of strathclyde
c          ja7.08
c          tel. 0141-548-4196
c
c date:    04/10/06
c
c version: 1.1 date: 04/10/2006
c modified: hugh p summers
c - first edition.
c
c version  : 1.2
c date     : 15-01-2007
c modified : Hugh Summers
c              - corrected metastable count for Ne+0.
c
c-----------------------------------------------------------------------
c-----------------------------------------------------------------------
      CHARACTER*80        CPTN_STACK(NDSTACK)
      INTEGER             IPTNL,       IZ0,         NCPTN_STACK, NDSTACK
```

9.157 xxmmul: Subroutine xxmmul from library adaslib

```
      SUBROUTINE XXMMUL( NR , NRC , NC , A , B , C )
C
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXMMUL *********************
C
C  PURPOSE: MULTIPLIES TWO MATRICES.
C
C  CALLING PROGRAM: GENERAL USE.
C
C  INPUT : (I*4)  NR      = NO. OF ROWS IN MATRICES 'A' AND 'C'.
C  INPUT : (I*4)  NRC     = NO. OF COLUMNS IN MATIX 'A' AND ROWS IN
C                           MATIRX 'B'.
C  INPUT : (I*4)  NC      = NO. OF COLUMNS IN MATRICES 'B' AND 'C'.
C  INPUT : (R*8)  A(,)    = FIRST MATRIX STORED AS A LINEAR VECTOR BY
C                           COLUMN.
C                           1ST DIMENSION: NR
C                           2ND DIMENSION: NRC
C  INPUT : (R*8)  B(,)    = SECOND MATRIX STORED AS A LINEAR VECTOR BY
C                           COLUMN.
C                           1ST DIMENSION: NRC
C                           2ND DIMENSION: NC
C
C  OUTPUT: (R*8)  C(,)    = RESULT MATRIX STORED AS LINEAR VECTOR BY
C                           COLUMN.
C                           1ST DIMENSION: NR
C                           2ND DIMENSION: NC
C                           DIMENSION: MXTERM
C
C          (I*4)  I       = LOOP INDEX.
C          (I*4)  J       = LOOP INDEX.
C          (I*4)  K       = LOOP INDEX.
C
C ROUTINES: NONE
C
C AUTHOR:  JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 5183
C
C DATE:    10/11/93
C
C UNIX-IDL PORT:
C
C AUTHOR:  WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C
C DATE:    22ND MAY 1996
C
C VERSION: 1.1                          DATE: 22-05-96
C MODIFIED: WILLIAM OSBORN
C               - FIRST VERSION. IBM VERSION NOT CHANGED
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             NC,          NR,          NRC
      REAL*8              A(NR,NRC),   B(NRC,NC),   C(NR,NC)
```

9.158 xxmnmx: Subroutine xxmnmx from library adaslib

```
      SUBROUTINE XXMNMX( LOGFIT , MAXDEG , TOLVAL ,
     &                   NIN    , XIN    , YIN    ,
     &                   NOUT   , XOUT   , YOUT   ,
     &                   NCOEF  , COEF ,
     &                   MINFO
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXMNMX *********************
C
C  PURPOSE:           TO EVALUATE THE TAYLOR COEFFICIENTS OF THE MINIMAX
C                     POLYNOMIAL.
C
C                     OUTPUTS 'NOUT' INTERPOLATED COORDINATES  USING THE
C                     MINIMAX EVALUATION, WHICH ARE EVENLY  SPACED ALONG
C                     THE X-AXIS COVERING THE X-VALUE RANGE  INPUT.   IF
C                     REQUIRED THE SPACING WILL BE EVENLY  SPACED  ALONG
C                     THE LOG10 TRANSFORMATION OF THE  X-AXIS  (IN  THIS
C                     CASE THE Y-VALUES ARE SIMILARLY TRANSFORMED).
C
C                     * IMPORTANT: 'NOUT' >= 'NIN' (NO. OF INPUT VALUES)
C
C                     MINFO(CHARACTER*80) ON OUTPUT CONTAINS INFORMATION
C                     REGARDING THE SLOPE OF THE CURVE AT THE BOUNDARIES
C                     OF THE EXPERIMENTAL REGION, AND AN ESTIMATE OF THE
C                     MAXIMUM ERROR IN THE MINIMAX INTERPOLATION.
C
C  CALLING PROGRAMS:  GENERAL USE
C
C  SUBROUTINE:
C
C  INPUT : (L*4)  LOGFIT  = .TRUE.  => PERFORM MINIMAX POLYNOMIAL EVAL-
C                                      UATION ON LOG10 TRANSFORMATIONS
C                                      OF THE INPUT X AND Y DATA.
C                           .FALSE  => PERFORM MINIMAX POLYNOMIAL EVAL-
C                                      UATION ON THE X AND Y VALUES AS
C                                      INPUT.
C  INPUT : (I*4)  MAXDEG  = MAXIMUM   POSSIBLE  DEGREE  OF   POLYNOMIAL
C                           ALLOWED IN THE MINIMAX FITTING (MUST BE< 26)
C                           (IF 'NOUT'<100 => MAX. DEGREES ALLOWED = 22)
C                           NOTE:
C                           'COEF()' MUST HAVE .GE. 'MAXDEG+1' ELEMENTS
C  INPUT : (R*8)  TOLVAL  = PARAMETER = FRACTIONAL TOLERANCE FOR ACCEPT-
C                                       ANCE OF DATA FITTED  BY  MINIMAX
C                                       POLYNOMIAL.  (IF IT EQUALS  ZERO
C                                       THEN RUNS TO MAX. DEGREE).
C
C  INPUT : (I*4)  NIN     = NUMBER OF INPUT KNOTS
C  INPUT : (R*8)  XIN()   = INPUT X-VALUES OF KNOTS
C  INPUT : (R*8)  YIN()   = INPUT Y-VALUES OF KNOTS
C
C  INPUT : (I*4)  NOUT    = NUMBER  OF  OUTPUT  VALUES  REQUIRED  TO  BE
C                           INTERPOLATED USING MINIMAX POLYNOMIAL EVAL-
C                           UATION - SPACED EVENLY BETWEEN THE MINIMUM
C                           AND MAXIMUM VALUES OF THE (TRANSFORMED) X-
C                           VALUES INPUT.
C  OUTPUT: (R*8)  XOUT()  = INITIALLY: ORDERED/(TRANSFORMED) 'XIN()'
C                                      (ASCENDING ORDER).
C                                      'NIN' VALUES
```

```
C                          OUTPUT   : X-VALUES FOR WHICH INTERPOLATION
C                                     IS CARRIED OUT.
C                                     'NOUT' VALUES
C  OUTPUT: (R*8)  YOUT()  = INITIALLY: ORDERED/(TRANSFORMED) 'YIN()'
C                                     (ORDERED ACCORDING TO 'XOUT()').
C                                     'NIN' VALUES
C                          OUTPUT   : INTERPOLATED Y-VALUES FOR THE
C                                     X-VALUES ('XOUT()' - OUTPUT)
C                                     'NOUT' VALUES
C
C  OUTPUT: (I*4)  NCOEF   = NUMBER OF MINIMAX FIT COEFFICIENTS
C  OUTPUT: (R*8)  COEF()  = MINIMAX COEFFICIENTS - ARRAY SIZE: MAXDEG+1
C
C  OUTPUT: (C*80) MINFO   = DIAGNOSTIC INFORMATION STRING
C
C          (I*4)  MAXOUT  = PARAMETER = MAXIMUM NUMBER OF OUTPUT X,Y CO-
C                                       ORDINATES THAT CAN BE INTERPOLA-
C                                       TED.
C          (I*4)  MAXTOL  = PARAMETER = POWER OF 10 WHICH REPRESENTS THE
C                                       MAXIMUM FRACTIONAL DIFFERENCE
C                                       ALLOWED BETWEEN ACTUAL (YDATA)
C                                       AND FITTED (YFIT) DATA.
C          (I*4)  MINTOL  = PARAMETER = POWER OF 10 WHICH REPRESENTS THE
C                                       MINIMUM FRACTIONAL DIFFERENCE
C                                       ALLOWED BETWEEN ACTUAL (YDATA)
C                                       AND FITTED (YFIT) DATA.
C
C          (I*4)  I4UNIT  = FUNCTION (SEE ROUTINE SECTION BELOW)
C          (I*4)  ILIMIT  = UPPER LIMIT  OF  THE  NUMBER  OF  DEGREES
C                           ALLOWED FOR MINIMAX POLYNOMIAL EVALUATION.
C          (I*4)  IDEG    = NUMBER OF DEGREES BEING USED FOR MINIMAX
C                           POLYNOMIAL EVALUATION.
C          (I*4)  NDEG    = NUMBER OF DEGREES FOR ACCEPTED MINIMAX FIT.
C          (I*4)  I1      = GENERAL ARRAY INDEX
C          (I*4)  I2      = GENERAL ARRAY INDEX
C
C          (R*8)  XMIN    = MINIMUM (TRANSFORMED) X-VALUE INPUT
C          (R*8)  XMAX    = MAXIMUM (TRANSFORMED) X-VALUE INPUT
C          (R*8)  REF     = OUTPUT FROM 'XXCHEB' = NEGATIVE IF MINIMAX
C                           FITTING PROCEDURE IS CYCLING.
C          (R*8)  SUM     = USED TO SUM UP POLYNOMIAL TERMS WHEN CALCUL-
C                           ATING VALUE OF 'Y' AT GIVEN 'X'.
C          (R*8)  YFIT    = FITTED Y-VALUE USING MINIMAX POLYNOMIAL
C          (R*8)  YDATA   = INPUT Y-VALUE FOR COMPARISON TO FITTED VALUE
C          (R*8)  DIFF    = FRACTIONAL DIFFERENCE BETWEEN ACTUAL (YDATA)
C                           AND  FITTED  (YFIT)  Y-VALUE   FROM  MINIMAX
C                           POLYNOMIAL FITTED.
C          (R*8)  BIG     = LARGEST FRACTIONAL ERROR BETWEEN ACTUAL AND
C                           FITTED Y-VALUES FOR POLYNOMIAL EVALUATION.
C          (R*8)  GRADL   = GRADIENT OF FITTED MINIMAX POLYNOMIAL CURVE
C                           AT LOWER BOUNDARY OF INPUT DATA.
C          (R*8)  GRADU   = GRADIENT OF FITTED MINIMAX POLYNOMIAL CURVE
C                           AT UPPER BOUNDARY OF INPUT DATA.
C          (R*8)  XSTEP   = SEPERATION   OF   OUTPUT  X-VALUES  TO   BE
C                           INTERPOLATED.
C          (R*8)  XVAL    = X-VALUES AT WHICH INTERPOLATION IS TAKING
C                           PLACE.
C
C          (C*6)  CFTYPE  = 'LOGFIT' IF 'LOGFIT=.TRUE.'
C                           'LINFIT' IF 'LOGFIT=.FALSE.'
C
```

```
C          (L*4)  LMFIT   = .TRUE.  => MINIMAX POLYNIOMIAL FOUND WHICH
C                                     IS WITHIN DESIRED TOLERANCE.
C                                     (SEE 'TOLVAL')
C                           .FALSE. => NO MINIMAX POLYNIOMIAL FOUND WITH
C                                     DESIRED TOLERANCE.
C                                     (SEE 'TOLVAL')
C
C          (I*4)  INDX()  = ASCENDING ORDER INDEX FOR INPUT X-VALUES
C
C          (R*8)  WRK()   = WORKING SPACE FOR ORDERING DATA
C
C
C NOTES:
C          THIS SUBROUTINE IS AN AMMENDED AND STRUCTURED VERSION OF  THE
C          SUBROUTINE  'MINIMAX'  WRITTEN BY  STEPHEN TURNER,  JET  30TH
C          JULY 1989. IT NO LONGER ORDERS THE INPUT COORDINATES DIRECTLY
C          BUT COPIES THE TO A SECOND ARRAY AND ORDERS THEM.   THEREFORE
C          IF THE INPUT COORDINATES ARE REQUIRED TO BE ORDERED THIS MUST
C          BE CARRIED OUT EXPLICITLY IN THE CALLING OPROGRAM/ROUTINE.
C          THE SUBROUTINE NO LONGERS REQUIRES THE INPUT CO-ORDINATES  IN
C          LOG10 FORM, THIS TRANSFORMATION CAN BE CARRIED OUT EXPLICITLY
C          BY THE SUBROUTINE.   THE  ROUTINE HAS BEEN WRITTEN TO  BE  OF
C          GENERAL RATHER THAN SPECIFIC USE.
C
C          IN THE ORIGINAL 'MINIMAX' SUBROUTINE THE MINIMUM NUMBER OF
C          DEGREES ACCEPTED FOR A FIT WAS 2, IN 'XXMNMX' IT IS 1.
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4UNIT     ADAS      FETCH UNIT NUMBER FOR OUTPUT OF MESSAGES
C          XXINDX     ADAS     GIVES INDICES OF SORTED ARRAY (ASCENDNG)
C          XXISRT     ADAS     RE-ORDERS ARRAY ACCORDING TO 'XXINDX'
C          XXCHEB     ADAS     MINIMAX POLYNOMIAL COEFT. EVALUATION
C
C AUTHOR:   PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C           K1/0/81
C           JET EXT. 4569
C
C DATE:     12/10/90
C
C UPDATE:  23/04/93 - PE BRIDEN - ADAS91: ADDED I4UNIT FUNCTION TO WRITE
C                                         STATEMENTS FOR SCREEN MESSAGES
C
C UPDATE:  24/05/93 - PE BRIDEN - ADAS91: CHANGED I4UNIT(0)-> I4UNIT(-1)
C
C UPDATE:  14/02/94 - PE BRIDEN - ADAS91: CORRECTED BUG WHEN CHECKING
C                                         TOLERANCE FOR LOGFIT - CHANGED
C                                         THE LINE:
C                                                  YDATA = YIN(I1)
C                                         TO:
C                                                  YDATA = YIN(INDX(I1))
C                                         (WOULD CAUSE PROBLEMS IF YIN
C                                          NOT ORDERED ON INPUT.)
C
C UPDATE:  14/02/94 - PE BRIDEN - ADAS91: RECODED SECTION USE TO TEST
C                                         TOLERANCE VALUE - NOW CHECKS
C                                         FOR DIVISION BY 0 AND USES
C                                         LOG10 TO STORES VALUES THUS
C                                         AVOIDING OVERFLOWS.
C                                         (BIG IS NOW STORED AS LOG10
```

```
C                                          VALUES INITIALLY)
C                                         INTRODUCED: MAXTOL & MINTOL
C                                                 - / -
C                                         STOP EXECUTING XXCHEB IF
C                                         CYCLING (CHECK REF VARIABLE)
C                                                 - / -
C                                         INITIALISED NDEG AND BIG TO
C                                         STOP ICA WARNING.
C
C UPDATE:  31/10/94 - PE BRIDEN - ADAS91: REPLACED CALL TO NAG ROUTINE
C                                         E02ACF WITH A CALL TO THE NEW
C                                         EQUIVALENT ADAS ROUTINE CALLED
C                                         XXCHEB (IDENTICAL ARGUMENTS).
C UPDATE:  17/1/95 - L JALOTA - IDL-ADAS : MODIFIED ERROR CHECKING
C    SECTION TO WRITE TO UNIT 0
C    NOT STDOUT WHICH INTERFERES
C    WITH PIPE COMMUNICATIONS.
C       :   6/3/95 - L JALOTA - IDL-ADAS : REPLACED CALL TO I4UNIT.
C
C UNIX-IDL PORT:
C
C VERSION: 1.2                          DATE: 10-11-95
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - INCREASED PARAMETER MAXOUT FROM 100 TO 120
C                 PRIMARILY TO PREVENT PROBLEMS WITH ADAS 506 WHERE
C                 101 POINTS ARE REQUIRED.
C
C VERSION  : 1.3
C DATE     : 10-04-2007
C MODIFIED : Allan Whiteford
C               - Modified documentation as part of automated
C   subroutine documentation preparation.
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      CHARACTER*80        MINFO
      INTEGER             MAXDEG,      NCOEF,       NIN,         NOUT
      LOGICAL             LOGFIT
      REAL*8              COEF(MAXDEG+1),           TOLVAL
      REAL*8              XIN(NIN),    XOUT(NOUT),  YIN(NIN)
      REAL*8              YOUT(NOUT)
```

9.159 xxname: Subroutine xxname from library adaslib

```
      SUBROUTINE XXNAME(REALNAME)

C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXNAME *********************
C
C PURPOSE: TO DETERMINE THE REAL NAME OF THE USER BY EXAMINING THE
C          SYSTEM /etc/passwd FILE. A C PROGRAM READS THE FILE.
C
C CALLING PROGRAM: GENERAL USE
C
C OUTPUT: (C*30)  REALNAME    = REAL NAME OF USER IF IT IS RECORDED
C                               OTHERWISE A DEFAULT STRING IS RETURNED
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          GETENV     SYSTEM    GETS USERNAME (8 LETTERS MAX)
C          XXSLEN     ADAS      FINDS NON BLANK POSITIONS IN STRING
C          PERS_F     ADAS      C ROUTINE TO INTEROGATE SYSTEM
C
C
C AUTHOR   : Martin O'Mullane
C DATE     : 11/08/98
C VERSION  : 1.1                          DATE: 11-08-98
C MODIFIED : Martin O'Mullane
C            FIRST VERSION
C
C VERSION: 1.1 DATE: 01-12-98
C MODIFIED: RICHARD MARTIN
C - PUT UNDER SCCS CONTROL
C
C VERSION: 1.2 DATE: 15-12-98
C MODIFIED: Martin O'Mullane
C - Replace USER with LOGNAME as location of username in the
C                environment variables. Linux, at least at JET, does not
C                assign USER. Other OS appear to set both.
C
C VERSION: 1.3 DATE: 19-01-06
C MODIFIED: Allan Whiteford
C - Changed test on REALNAME to reflect changes in
C                underlying C code. Also moved removal of last
C                character to after 'Who produced this file' is
C                possibly set.
C
C VERSION: 1.4 DATE: 06-12-06
C MODIFIED: Allan Whiteford
C - Updated to allow for USERIDs > 8 characters (now set
C                to 20).
C
C VERSION: 1.5 DATE: 05-07-07
C MODIFIED: Allan Whiteford
C - Add on CHAR(0) to username as C style string
C                terminator rather than '\0'
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      EXTERNAL      PERS_F
C-----------------------------------------------------------------------
```

```
      CHARACTER*30        REALNAME
```

9.160 xxnbaf: Subroutine xxnbaf from library adaslib

```
      SUBROUTINE XXNBAF (M, NCAP7 , X, Y, W, LAMDA , B, A, DIAG, C ,
     &                   SS , IFAIL )
C
C
C-----------------------------------------------------------------------
C
C  PURPOSE: Determines a least-square cubic spline approximation s(x)
C           to the set of data points (x_r, y_r) with weights w_r,
C     for r=1,2,...,m.
C
C           The value of NCAP7 = ncap+7, where ncap is the number of
C     intervals of the spline (number of interior knots + 1),
C           and the values of the knots LAMDA(5), LAMDA(6), ...,
C     LAMDA(NCAP7-4), interior to the data interval, are
C     prescribed by the user.
C
C     s has the property that it minimises ss, the sum of the
C     squares of the weighted residuals eps(r)
C
C       eps(r) = w(r) * ( s(x(r)) - y(r) ).
C
C     The procedure produces the minimising value of ss and
C     the coefficients c(1), c(2),...,c(q), where q=ncap+3=NCAP7-4,
C     in the B-spline representation
C
C s(x) = c(1)*N1(x) + c(2)*N2(x) + ... + c(q)*Nq(x) .
C
C     Here Ni(x) (i=1,2,...,q) denotes the normalised B-spline
C     of degree 3 defined upon the knots lamda(i-4), lamda(i-3),
C     lamda(i-2), lamda(i-1), and lamda(i).
C
C  CALLING PROGRAM: VARIOUS
C
C  SUBROUTINE:
C
C  INPUT: (I*4) M            = The number of data points.
C                              CONSTRAINT: M >= MDIST >= 4, where
C                              MDIST is the number of distinct x
C                              values in the data.
C
C  INPUT: (I*4) NCAP7        = NBAR+7, where NBAR is the number of
C                              intervals of the spline (number
C                              of interior knots +1, i.e. the knots
C                              strictly in the range X(1) to X(M))
C                              over which the spline is defined.
C                              CONSTRAINT: 8<= NCAP7 <= MDIST+4,
C                              where MDIST is the number of distinct
C                              x values in the data.
C
C  INPUT:  (R*8) X()         = The values x_r of the independent variable
C                              (abscissa), for r=1,2,...,m.
C                              CONSTRAINT:
C                                 X(1)<=X(2)<=...<=X(M)
C
C  INPUT:  (R*8) Y()         = The values y_r of the dependent variable
C                              (ordinate), for r=1,2,...,m.
C
C  INPUT:  (R*8) W()         = The values w_r of the weights,
C                              for r=1,2,...,m.
```

```
C
C  INPUT:  (R*8) LAMDA()     = LAMDA(i) must be set to the (i-4)th
C                               (interior) knot, i=5,6,...,nbar+3.
C                               CONSTRAINT:
C                                  X(1) < LAMDA(5) <= LAMDA(6) ... <=
C                                        ... <=LAMDA(NCAP7-4) < X(M) .
C
C  INPUT:  (I*8) IFAIL       = 0 : stop if any error
C                            = 1 : continue if non-fatal error.
C
C  OUTPUT: (R*8) LAMDA()     = Input values are unchanged, and
C                               LAMDA(i), for i=1,2,3,4,NCAP7-3,
C                               NCAP7-2,NCAP7-1,NCAP7 contains the
C                               additional exterior knots introduced by
C                               the routine.
C
C
C  OUTPUT: (R*8) C()         = The coefficients of the B-spline N_i(x),
C                               for i=1,2,...,nbar+3. The remaining
C                               elements (from NBAR+4 to NBAR+7) are not
C                               used.
C
C  OUTPUT: (R*8) SS          = The residual sum of sqaures
C
C  OUTPUT: (I*4) IFAIL       = 0 : no error detected
C                            = 1 : the knots fail to satisfy the condition
C                                  X(1) < LAMDA(5) <= LAMDA(6) <=...
C                                    <= LAMDA(NCAP7-4) < X(M)
C                            = 2 : The weights are not strictly positive
C                            = 3 : The values of X(R), R=1,M are not in
C                                  non-decreasing order.
C                            = 4 : NCAP7 < 8 (so that the number of
C                                  interior knots is negative) or
C                                  NCAP7 > MDIST + 4, where MDIST is the
C                                  number of distinct x values in the data
C                                  (so that there cannot be unique solution).
C                            = 5 : The conditions specified by Schoenberg
C                                  and Whitney fail to hold for at least
C                                  one subset of the distinct data abscissae.
C                                  That is, there is no subset of NCAP7-4
C                                  strictly increasing values,
C                                  X(R(1)), X(R(2)),..., X(R(NCAP7-4)),
C                                  among theabscissae such that
C
C                                  X(R(1)) < LAMDA(1) < X(R(5))
C                                  X(R(2)) < LAMDA(2) < X(R(6))
C                                  ...
C                                  X(R(NCAP7-8))<LAMDA(NCAP7-8)<X(R(NCAP7-4)).
C
C                                  This means that there is no unique
C                                  solution: there are regions containing
C                                  too many knots compared with the
C                                  number of data points.
C
C          (R*8)  B()         = Set of distinct data abscissae
C
C          (R*8)  WORK2()     = WORKSPACE
C
C          (I*4)  J           = GENERAL INDEX
C          (I*4)  I           = GENERAL INDEX
C          (I*4)  R           = GENERAL INDEX
```

```
C          (I*4)  II         = GENERAL INDEX
C          (R*8)  BI         = GENERAL REAL
C          (R*8)  XI         = GENERAL REAL
C
C  ROUTINES:  NONE
C
C  AUTHORS: Alessandro C. Lanzafame, University of Strathclyde
C
C  REFERENCE: Cox, M.G. and Hayes, J.G. "Curve fitting: A Guide and
C             Suite of Algorithms for the Non-specialist User."
C             Report NAC26, National Physical Laboratory, Middlessex,
C             1973.
C
C  DATE:    12 January 1995
C
C  VERSIION: 1.0a
C  Alessandro Lanzafame, 12 January 1995.
C  Directly derived from Algol text.
C  (Error in passing woking variables)
C
C  VERSION 1.0b
C  Alessandro Lanzafame, 15 January 1995.
C  DIAG(1:NCAP7-4) absorbed in matrix A(1:NCAP7-4,1).
C  Matrix A(1:NCAP7-4,2:4) becomes A(1:NCAP7-4,1:4). This is to easy
C  the passing of workspaces.
C  WORK1 identified with B. WORK2 with A.
C  Corrected index error in checking remaining Schoennber-Whitney
C  conditions.
C  COMPILING BUT NOT WORKING.
C
C  VERSION 1.0c
C  Alessandro Lanzafame, 15 January 1995.
C  Knots shifted as in original Algol program text.
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 22-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - PUT UNDER SCCS CONTROL
C
C VERSION: 1.2                          DATE: 06-07-2004
C MODIFIED: Allan Whiteford
C               - Changed name from dxnbaf to xxnbaf.
C
C VERSION  : 1.3 DATE: 10-04-2007
C MODIFIED : Allan Whiteford
C               - Modified documentation as part of automated
C   subroutine documentation preparation.
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IFAIL,       M,           NCAP7
      REAL*8              A(1:NCAP7-4,2:4),         B(M)
      REAL*8              C(NCAP7),    DIAG(1:NCAP7-4)
      REAL*8              LAMDA(-3:NCAP7-4),        SS,          W(M)
      REAL*8              X(M),        Y(M)
```

9.161 xxnbbf: Subroutine xxnbbf from library adaslib

```
      SUBROUTINE XXNBBF( NCAP7 , LAMDA , C , X ,S , IFAIL )
C
C
C-----------------------------------------------------------------------
C
C  PURPOSE: Evaluates a cubic spline from its B-spline representation
C
C  CALLING PROGRAM: VARIOUS
C
C  SUBROUTINE:
C
C  INPUT: (I*4) NCAP7        = NBAR+7, where NBAR is the number of
C                              intervals (one greater than the number
C                              of interior knots, i.e. the knots
C                              strictly in the range LAMDA(4) to
C                              LAMDA(NCAP+4)) over which the spline
C                              is defined.
C                              CONSTRAINT: NCAP7 >= 8
C
C  INPUT:  (R*8) LAMDA()     = Values of the complete set of knots
C                              LAMDA(J), J=1,NBAR+7.
C                              CONSTRAINT: Must be in non-decreasing
C                              order with LAMDA(NCAP7-3) > LAMDA(4).
C
C  INPUT:  (R*8) C()         = The coefficients of the B-spline N_i(x),
C                              for i=1,2,...,nbar+3. The remaining
C                              elements (from NBAR+4 to NBAR+7) are not
C                              used.
C
C  INPUT:  (R*8) X           = The argument x at which the cubic spline
C                              is to be evaluated.
C                              CONSTRAINT:
C                                 LAMDA(4) <= X <= LAMDA(NCAP7-3)
C  INPUT:  (I*8) IFAIL       = 0 : stop if any error
C                            = 1 : continue if non-fatal error.
C
C  OUTPUT: (R*8) S           = The value of the spline, s(x)
C
C  OUTPUT: (I*4) IFAIL       = 0 : no error detected
C                            = 1 : X does not satisfy
C                                  LAMDA(4) <= X <= LAMDA(NCAP7-3)
C                            = 2 : NCAP7 < 8
C
C
C          (I*4)  KL          = GENERAL INDEX
C          (I*4)  KU          = GENERAL INDEX
C          (I*4)  K           = GENERAL INDEX
C          (I*4)  J           = GENERAL INDEX
C          (R*8)  E1          = GENERAL REAL
C          (R*8)  E2          = GENERAL REAL
C
C  ROUTINES:  NONE
C
C  AUTHORS: H. P. Summers and A. C. Lanzafame, University of Strathclyde
C
C  DATE:    11 January 1995
C
C UNIX-IDL PORT:
C
```

```
C VERSION: 1.1                          DATE: 22-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - PUT UNDER SCCS CONTROL
C
C VERSION: 1.2                          DATE: 06-07-2004
C MODIFIED: Allan Whiteford
C               - Changed name from dxnbbf to xxnbbf
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IFAIL,       NCAP7
      REAL*8              C(NCAP7),    LAMDA(NCAP7),             S
      REAL*8              X
```

9.162 xxopcs: Subroutine xxopcs from library adaslib

```
      subroutine xxopcs(unit,file,type)


C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXOPCS *********************
C
C  PURPOSE: Open a file on a given unit but when a read is done from
C           that unit, only give the contents back in a given case.
C
C  CALLING PROGRAM: GENERAL USE.
C
C  INPUT    : (I*4)   UNIT   = Unit to open the file on
C  INPUT    : (C*(*)) FILE   = Input Filename
C  INPUT    : (C*2)   TYPE = Type of case to convert to:
C                       'UC' -> Convert to Upper Case
C                       'LC' -> Convert to Lower Case
C                       Anything else -> No conversion
C
C  OUTPUT   : NONE
C
C  ROUTINES :
C            ROUTINE    SOURCE    BRIEF DESCRIPTION
C            ------------------------------------------------------------
C            XXCASE     ADAS      CONVERT STRING TO UPPER OF LOWER CASE
C
C  AUTHOR   : Allan Whiteford,
C             University of Strathclyde
C
C  VERSION  : 1.1                          DATE: 05/09/2001
C  MODIFIED : Allan Whiteford
C             First version.
C
C-----------------------------------------------------------------------
      CHARACTER*(*)       FILE
      CHARACTER*2         TYPE
      INTEGER             UNIT
```

9.163 xxopen: Subroutine xxopen from library adaslib

```
       SUBROUTINE XXOPEN( IUNIT, DSFULL , LEXIST )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXOPEN *********************
C
C  PURPOSE: TO INQUIRE & OPEN INPUT DATA FILE & ALLOCATE TO UNIT 'IUNIT'
C           (READ ONLY)  - IF IT DOES NOT EXISTS A MESSAGE IS  SENT  TO
C           THE SCREEN AND LEXIST IS RETURNED AS FALSE.
C
C  CALLING PROGRAM:  GENERAL USE
C
C  SUBROUTINE:
C
C  INPUT :    (I*4)  IUNIT   = UNIT TO WHICH INPUT FILE IS ALLOCATED
C  INPUT :    (C*(*))DSFULL  = FULL INPUT DATA SET NAME (INCL. USERID)
C                              IN FORM FOR DYNAMIC ALLOCATION.
C  OUTPUT:    (L*4)  LEXIST  = .TRUE.  => DATA SETS EXISTS AND IS OPEN
C                            = .FALSE. => DATA SET DOES NOT EXIST
C
C             (I*4)  I4UNIT  = FUNCTION (SEE ROUTINE SECTION BELOW)
C             (C*1)  BSLASH  = '/' - MUST BE FIRST 'DSFULL'CHARACTER
C NOTE:
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4UNIT     ADAS      FETCH UNIT NUMBER FOR OUTPUT OF MESSAGES
C          XXTERM     ADAS      TERMINATES ADAS PROGRAM WITH MESSAGE
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/37
C          JET EXT. 2520
C
C DATE:    04/03/91 - ADAS91 VERSION (INCLUDES INQUIRE)
C
C UPDATE:  07/08/91 - PE BRIDEN: ADDED ERROR HANDLING FOR OPEN ERROR.
C
C UPDATE:  23/04/93 - PE BRIDEN - ADAS91: ADDED I4UNIT FUNCTION TO WRITE
C                                         STATEMENTS FOR SCREEN MESSAGES
C
C UPDATE:  24/05/93 - PE BRIDEN - ADAS91: CHANGED I4UNIT(0)-> I4UNIT(-1)
C
C-----------------------------------------------------------------------
      CHARACTER*(*)       DSFULL
      INTEGER             IUNIT
      LOGICAL             LEXIST
```

9.164 xxordr: Subroutine xxordr from library adaslib

```
      subroutine xxordr(n, lup, xa, ia, iinva)

c-----------------------------------------------------------------------
c
c  ****************** fortran77 subroutine: xxordr ******************
c
c  purpose: sorts a real*8 array xa. This is a bubble sort designed for
c           small arrays.
c
c
c  calling program:  general use
c
c
c  subroutine:
c
c  input       :  (i*4)  n       = size of input arrays
c  input       :  (l*4)  lup     = .true. sort in ascending order
c                                  .false. descending order
c  input/output:  (r*8)  xa      = array to be sorted
c  output      :  (i*4)  ia      = original index of sorted xa members
c  output      :  (i*4)  iinva   = inverse of ia
c
c  routines: none
c
c  notes: based on nsort.  subroutine sets ia(i)=i before sorting
c
c  author:  Susan Turnbull
c
c  date:    22/08/01
c
c  version: 1.1
c
c-----------------------------------------------------------------------
      INTEGER             IA(*),       IINVA(*),    N
      LOGICAL             LUP
      REAL*8              XA(*)
```

9.165 xxpars: Subroutine xxpars from library adaslib

```
       subroutine xxpars( ndmet  , strng1 , npt    , bwnoa   , lseta   ,
     &                    prtwta , cprta  , ifail  , itype)
c
c-----------------------------------------------------------------------
c
c  ****************** fortran77 subroutine: xxpars *********************
c
c  purpose:  to analyse the tail character string of the first line of
c            a specific ion file into binding wave numbers for different
c            parents and statistical weights for the parents.
c
c            unified version of b9pars, bapars & g5pars and a
c            replacement for these subroutines.
c
c  calling program: various
c
c
c  subroutine:
c
c  input : (i*4)  ndmet    =  maximum number of metastables allowed
c  input : (c*(*))strng1   =  string to be parsed
c
c  output: (i*4)  npt      =  number of binding wave numbers detected
c  output: (l*4)  lseta()  =  .true.  - parent term set for this w.no.
c                             .false. - parent term not set for w.no.
c  output: (l*4)  lfnd     =  .true.  - l quantum number present in
c                                       string
c                             .false. - no l quantum number detected
c  output: (r*8)  bwnoa()  =  binding wave numbers
c  output: (r*8)  prtwta() =  parent statistical weights
c  output: (c*(*))cprta()  =  parent name in brackets
c  output: (i*4)  ifail    =  0 - subroutine concludes correctly
c                             1 - fault detected in subroutine
c                             2 - single ionisation potential detected
c  output: (i*4)  itype    =  resolution of parent metastables
c                             1 - ls resolved
c                             2 - lsj resolved
c                             3 - arbitrary resolution
c
c          (i*4)  maxwrd   =  maximum number of words sought initially
c                             initially, finally number actually found
c          (i*4)  nfirst   =  first word to be extracted from string
c          (i*4)  ifirst() =  index of first char. of word () in string
c          (i*4)  ilast()  =  index of last  char. of word () in string
c          (i*4)  iwords   =  number of words found in string
c          (i*4)  iabt     =  failure number from r8fctn
c          (i*4)  nchar    =  number of characters in substring
c          (i*4)  i        =  general use
c          (i*4)  j        =  general use
c          (i*4)  k        =  general use
c          (i*4)  ic       =  general use
c          (i*4)  itp      =  flag for incompatible types
c          (i*4)  ityp     =  copy of current itype
c          (i*4)  kmrk     =  position marker in the string for parent
c                             l quantum number
c          (i*4)  itypea() =  resolution of each parent
c          (r*8)  twta()   =  (2L+1) value for parent L quantum number
c          (c*1)  ctrma()  =  parent L quantum number letter set
c                             (inclusive convention for 'l'=j in set of
```

```
c                              character values for 'l' and extended
c                              ctrma, twta vectors)
c routines:
c          routine    source    brief description
c          ------------------------------------------------------------
c          i4unit     adas      fetch unit number for output of messages
c          r8fctn     adas      converts from character to real variable
c          i4fctn     adas      converts from char. to integer  variable
c          xxword     adas      parses a string into separate words
c                               for ' ()<>{}' delimiters
c          xxslen     adas      finds the length of a string excluding
c                               leading and trailing blanks
c          xxrmve     adas      removes a character from a string
c          xxcase     adas      change string to upper or lower case
C
c
c author:  hp summers
c          JA7.08, University of Strathclyde
c          Tel: 0141-548-4196
c
c date:    30/01/03
c
c update:  22/11/04 - hp summers - corrected error in write back of
c                                  cprta strings for the unified itype
c
c update:  17/05/07 - ad whiteford - Updated comments as part of
C                                    subroutine documentation
C                                    procedure.
c
c-----------------------------------------------------------------------
      CHARACTER*(*)       CPRTA(NDMET),             STRNG1
      INTEGER             IFAIL,       ITYPE,       NDMET,       NPT
      LOGICAL             LSETA(NDMET)
      REAL*8              BWNOA(NDMET),             PRTWTA(NDMET)
```

9.166 xxpint: Subroutine xxpint from library adaslib

```
      subroutine xxpint(ndim, xa, ya, x, y)

C-----------------------------------------------------------------------
C
C  ******************** FORTRAN77 FUNCTION: XXPINT **********************
C
C  PURPOSE:  Order ndim polynominal interpolation.
C
C  CALLING PROGRAM: adas408
C
C  FUNCTION:
C
C  input : (i*4)  ndim     = number of enteries in input arrays.
C  input : (r*8)  xa       = independent variable.
C  input : (r*8)  ya       = dependent variable.
C  input : (r*8)  x        = use requested x value.
C  output: (r*8)  y        = interpolated y value.
C
C  NOTES: Assume the starting value for y is ya(2).
C         Based on polint.for of Numerical Recipes.
C
C  ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          i4unit     ADAS      Fetch unit number for message output
C
C
C  VERSION  : 1.1
C  DATE     : 23-07-2003
C  MODIFIED : Martin O'Mullane
C              - First version.
C
C-----------------------------------------------------------------------
      INTEGER             NDIM
      REAL*8              X,           XA(NDIM),    Y
      REAL*8              YA(NDIM)
```

9.167 xxpixv: Subroutine xxpixv from library adaslib

```
       subroutine xxpixv( ndwvl  , ndpix  , fcrit  ,
     &                    nwvl   , npix   , wvmin  , wvmax  ,
     &                    iwvrg  , cpixmx ,
     &                    wvl    , tev    , amssno , pec    ,
     &                    cpixa  , ind1   , ind2
     &                  )

c-----------------------------------------------------------------------
c
c  ****************** fortran77 subroutine: xxpixv *********************
c
c  purpose:  distribute Doppler broadened line emission into pixel range
c
c  calling program: hapecf, adas316
c
c
c  subroutine:
c
c  input : (i*4)  ndwvl   = maximum number of wavelength intervals
c  input : (i*4)  ndpix   = maximum number of pixels per wvln. interval
c  input : (r*8)  fcrit   = pixel counts for the selected line below
c                           this fraction of the largest pixel count are
c                           discounted.
c
c  input : (i*4)  nwvl    = number of wavelength intervals
c  input : (i*4)  npix()  = number of pixels assigned to wavelength interval
c  input : (r*8)  wvmin() = lower limit of wavelength interval (ang)
c  input : (r*8)  wvmax() = upper limit of wavelength interval (ang)
c
c  input : (i*4)  iwvrg   = index of wavelength range in which line lies
c  input : (r*8)  cpixmx  = largest pixel count currently found
c                           for the wavelength range
c
c  input : (r*8)  wvl     = input line wavelength for test(ang)
c  input : (r*8)  tev     = electron temperature (eV)
c  input : (r*8)  amssno  = atomic mass number
c  input : (r*8)  pec     = photon emissivity coefficient for line
c
c  output: (r*8)  cpixa() = counts in each pixel for the line
c  output: (r*8)  ind1    = first pixel with non-negligible count
c  output: (r*8)  ind2    = last pixel with non-negligible count
c
c routines:
c          routine    source    brief description
c          ------------------------------------------------------------
c          i4unit     adas      fetch unit number for output of messages
c          r8erfc     adas      returns erfc(x) function value
c
c author:  Hugh Summers, University of Strathclyde
c          JA7.08
c          tel. 0141-548-4196
c
c date:    15/01/02
c
c version  : 1.1
c date     : 15-01-2002
c modified : H P Summers
c             - first version.
c
```

```
c version  : 1.2
c date     : 18-06-2007
c modified : H P Summers
c             - corrected error in ind1 & ind2 return.
c             - renamed to xxpixv.
c
c-----------------------------------------------------------------------
c-----------------------------------------------------------------------
      INTEGER             IND1,         IND2,         IWVRG,        NDPIX
      INTEGER             NDWVL,        NPIX(NDWVL), NWVL
      REAL*8              AMSSNO,       CPIXA(NDPIX),              CPIXMX
      REAL*8              FCRIT,        PEC,          TEV,          WVL
      REAL*8              WVMAX(NDWVL),               WVMIN(NDWVL)
```

9.168 xxprs1: Subroutine xxprs1 from library adaslib

```
       subroutine xxprs1(ndmet,string,wno,cpl,npt,ipla,zpla,ifail)

C-----------------------------------------------------------------------
C
C  ****************** fortran77 subroutine: xxprs1 *********************
C
C  purpose:  to analyse the tail character string of an level data line
C            of an adf04 specific ion file into wave-number and sets of
C            (parent identifier, effective zeta for the parent) pairs.
C
C            unified version of baprs1, b9prs1, bbprs1, g5prs1 which is
C            a replacement for these subroutines
C
C  calling program: various
C
C  notes: detect  -  level wave number which preceeds first '{'
C                 -  sets of   parent index contained in '{.}'
C                              followed by effective zeta
C         nb. 'x' as first parent assignment means exclude ionisation
C             from this level.
C             no parent assignment means take lowest parent with
C             zeta =1.
C             lowest parent but no zeta means take zeta =1.
C             if there is more than one parent then zeta's must be in.
C
C
C  subroutine:
C
C  input : (i*4)  ndmet    =  maximum number of parents
C  input : (c*(*))string   =  string to be parsed
C
C  output: (r*8)  wno      =  excitation wave number of level relative
C                             to lowest parent
C  output: (c*1)  cpl      =  lead parent for ionisation  or 'x'
C  output: (i*4)  npt      =  number of parents detected
C  output: (i*4)  ipla()   =  parent indices.
C  output: (r*8)  zpla()   =  effective zeta for parent ipla()
C  output: (i*4)  ifail    =  0 - subroutine concludes correctly
C                             1 - fault detected in subroutine
C                             2 - single ionisation potential detected
C
C          (i*4)  maxwrd   =  maximum number of words sought initially
C                             initially, finally number actually found
C          (i*4)  nfirst   =  first word to be extracted from string
C          (i*4)  ifirst() =  index of first char. of word () in string
C          (i*4)  ilast()  =  index of last  char. of word () in string
C          (i*4)  iwords   =  number of words found in string
C
C          (l*4)  lset     =  .true.  -  wave number part set
C                             .false. -  wave number part not set
C          (l*4)  lwno     =  .true.  -  in the wave number part
C                             .false. -  not in the wave number part
C          (l*4)  lprnt    =  .true.  -  in a parent specifier
C                             .false. -  not in a parent specifier
C          (l*4)  lzeta    =  .true.  -  in a zeta specifier
C                             .false. -  not in a zeta specifier
C          (i*4)  ic       =  general use
C          (i*4)  iabt     =  failure number from r8fctn
C          (i*4)  nchar    =  number of characters in substring
```

```
C          (c*15) sstrng   =  isolated substring
C
C routines:
C          routine    source    brief description
C          ------------------------------------------------------------
C          i4unit     adas      fetch unit number for output of messages
C          r8fctn     adas      converts from character to real variable
C          i4fctn     adas      converts from char. to integer  variable
C          xxword     adas      parses a string into separate words
C                               for ' ()<>{}' delimiters
C
C AUTHOR:  HP Summers
C          JA7.08, University of Strathclyde
C          Tel: 0141-548-4196
C
C DATE:    04/12/02
C
C UPDATE:   :
C
C-----------------------------------------------------------------------
      CHARACTER           CPL
      CHARACTER*(*)       STRING
      INTEGER             IFAIL,       IPLA(NDMET), NDMET,       NPT
      REAL*8              WNO,         ZPLA(NDMET)
```

9.169 xxprs3: Subroutine xxprs3 from library adaslib

```
       subroutine xxprs3( string , ia_dim     , iz0   , iz    ,
     &                    ia     , ia_vlnce
     &                  )
c-----------------------------------------------------------------------
c
c  ****************** fortran77 subroutine: xxprs3 *********************
c
c  purpose:  To analyse a configuration character string in Standard
c            form into a  integer array of occupation numbers in the
c            normal collating order.
c
c            unified version of bbprs3 & g5prs3 and a replacement for
c            these subroutines

c  calling program: various
c
c  notes:   1. the normal collating order is 1s,2s,2p,3s,3p,3d etc.
c           2. a strategy for deciding the occupancy of unspecified
c              inner shells is implemented using the nuclear and ion
c              charge.  The strategy may be switched off.
c           3. opens a central ADAS adf00 file for an element on
c              unit 10.
c
c
c  subroutine:
c
c  input : (c*(*))string   =  string to be parsed
c  input : (i*4)  ia_dim   =  dimensionality of occupancy vector ia
c  input : (i*4)  iz0      =  nuclear charge
c  input : (i*4)  iz       =  ion charge
c
c  output: (i*4)  ia()     =  set of occupation numbers in standard
c                             collating list order
c  output: (i*4)  ia_vlnce =  outer occupied shell index in collating
c                             list
c
c  routines:
c           routine    source    brief description
c           ------------------------------------------------------------
c           i4unit     adas      fetch unit number for output of messages
c           xxopen     adas      inquire & open a data file on a unit
c           xxword     adas      parses a string into separate words
c           xxrmve     adas      removes occurrences of a char. in string
c           xxcase     adas      changes a string to upper or lower case
c           xxslen     adas      finds the length of a string excluding
c                                leading and trailing blanks
c           xfesym     adas      fetch the chemical symbol of an element
c
c
c author:  H. P. Summers
c          JA7.08, University of Strathclyde
c          Tel: 0141-548-4196 4941
c
c date:    16/12/02
c
c update:
c
c-----------------------------------------------------------------------
      CHARACTER*(*)       STRING
```

```
      INTEGER             IA(IA_DIM),  IA_DIM,      IA_VLNCE,    IZ
      INTEGER             IZ0
```

9.170 xxr8sort: Subroutine xxr8sort from library adaslib

```
      SUBROUTINE XXR8SORT(N, LUP, XA, IA)

C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXR8SORT ******************
C
C  PURPOSE: Sorts a REAL*8 array xa and its index array. This is a
C           bubble sort designed for small arrays.
C
C
C  CALLING PROGRAM:  General use
C
C
C  SUBROUTINE:
C
C  INPUT       :  (I*4)  N   = Size of input arrays
C  INPUT       :  (L*4)  LUP = .TRUE. sort in ascending order
C                              .FALSE. descending order
C  INPUT/OUPUT :  (R*8)  XA  = Array to be sorted
C  INPUT/OUPUT :  (I*4)  IA  = Original index of sorted XA
C
C  ROUTINES: NONE
C
C  NOTES: Based on nsort
C
C  AUTHOR:  Martin O'Mullane
C           K1/1/52
C
C  DATE:    19/07/99
C
C  VERSION: 1.1
C
C
C-----------------------------------------------------------------------

C-----------------------------------------------------------------------
      INTEGER             IA(*),       N
      LOGICAL             LUP
      REAL*8              XA(*)
```

9.171 xxr8ss: Subroutine xxr8ss from library adaslib

```
      subroutine xxr8ss(n, lup, ia, itag)

C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXR8SS ********************
C
C  PURPOSE: Sorts an integer array xa and its index array. This is a
C           bubble sort designed for small arrays.
C
C
C  CALLING PROGRAM:  General use
C
C
C  SUBROUTINE:
C
C  INPUT       :  (I*4)  N     = Size of input arrays
C  INPUT       :  (L*4)  LUP   = .TRUE. sort in ascending order
C                                .FALSE. descending order
C  INPUT/OUPUT :  (I*4)  IA    = Array to be sorted
C  INPUT/OUPUT :  (I*4)  ITAG  = Original index of sorted XA
C
C  ROUTINES    : NONE
C
C  NOTES       : Shell sort from Numerical Receipies.
C
C  AUTHOR      : Martin O'Mullane
C
C  DATE        : 7-02-2000
C
C  VERSION     : 1.1
C
C
C-----------------------------------------------------------------------
      INTEGER             ITAG(*),     N
      LOGICAL             LUP
      REAL*8              IA(*)
```

9.172 xxrams: Subroutine xxrams from library adaslib

```
      SUBROUTINE XXRAMS( VALUE )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXRAMS  *********************
C
C  PURPOSE: ADAS ROUTINE - SETS UP and interrogates a value for
C           the receiver  mass.
C
C              VALUE ON INPUT   =>   VALUE ON OUTPUT
C
C                     ?                 returns value
C                 <BLANK>               returns value
C                     *                 returns 'UNSET'
C                 <OTHER>               sets value
C
C
C  CALLING PROGRAM: GENERAL USE - designed for use with R8CONST
C
C  SUBROUTINE:
C
C   I/O   : (C*7)  VALUE   = VALUE UNDER WHICH ADAS DATA IS STORED
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C
C NOTE:
C          TO CHECK CURRENT RECEIVER MASS CALL XXRAMS WITH
C          ? AS INPUT.
C
C AUTHOR:  Martin O'Mullane
C
C DATE:    01/04/99
C
C UPDATE:
C
C-----------------------------------------------------------------------
      CHARACTER*(*)       VALUE
```

9.173 xxrate: Subroutine xxrate from library adaslib

```
      SUBROUTINE XXRATE( NDTRN  , NDTEM  , NDLEV ,
     &                   ICNT   , MAXT   ,
     &                   XJA    , ER     , TEMP  ,
     &                   I1A    , I2A    ,
     &                   RATE1  , DRATE1
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXRATE *********************
C
C  PURPOSE: TO CALCULATE THE EXCITATION AND DE-EXCITATION RATE COEFFICI-
C           ENTS FOR A SET OF INPUT TEMPERATURES(rydberg) & TRANSITIONS.
C           VALUES RETURNED ASSUMING UNIT GAMMA VALUES (I.E. GAMMA = 1)
C
C  CALLING PROGRAM:  GENERAL USE
C
C  SUBROUTINE:
C
C  INPUT :  (I*4)  NDTRN   = MAXIMUM NUMBER OF TRANSITIONS ALLOWED
C  INPUT :  (I*4)  NDTEM   = MAXIMUM NUMBER OF TEMPERATURES ALLOWED
C  INPUT :  (I*4)  NDLEV   = MAXIMUM NUMBER OF ENERGY LEVELS ALLOWED
C
C  INPUT :  (I*4)  ICNT    = NUMBER OF TRANSITIONS
C  INPUT :  (I*4)  MAXT    = NUMBER OF ISPF INPUT TEMPERATURES
C
C  INPUT :  (R*8)  XJA()   = QUANTUM NUMBER (J-VALUE) FOR GIVEN LEVEL.
C                            NOTE: (2*XJA)+1 = STATISTICAL WEIGHT
C  INPUT :  (R*8)  ER()    = ENERGY RELATIVE TO LEVEL 1 (RYDBERGS)
C                            DIMENSION ENERGY LEVEL.
C  INPUT :  (R*8)  TEMP()  = ISPF READ TEMPERATURES (KELVIN)
C
C
C  INPUT :  (I*4)  I1A()   = TRANSITION: LOWER ENERGY LEVEL INDEX
C  INPUT :  (I*4)  I2A()   = TRANSITION: UPPER ENERGY LEVEL INDEX
C
C  OUTPUT:  (R*8)  RATE1(,) = UNIT GAMMA EXCITATION RATE COEFFS(cm**3/s)
C                             1st DIMENSION: TEMPERATURE INDEX
C                             2nd DIMENSION: TRANSITION INDEX
C  OUTPUT:  (R*8)  DRATE1(,)= UNIT GAMMA DE-EXCIT'N RATE COEFFS(cm**3/s)
C                             1st DIMENSION: TEMPERATURE INDEX
C                             2nd DIMENSION: TRANSITION INDEX
C
C           (I*4)  NLTEM    = PARAMETER = MUST BE >= 'NDTEM'
C
C           (R*8)  TK2ATE   = PARAMETER = EQUATION CONSTANT = 1.5789D+05
C           (R*8)  R2GAM    = PARAMETER = EQUATION CONSTANT = 2.17161D-08
C
C           (I*4)  IC       = TRANSITION ARRAY INDEX
C           (I*4)  IT       = TEMPERATURE ARRAY INDEX
C
C           (R*8)  EUPPER   = SELECTED TRANSITION - UPPER ENERGY
C                             LEVEL RELATIVE TO INDEX LEVEL 1 (Rydbergs)
C           (R*8)  ELOWER   = SELECTED TRANSITION - LOWER ENERGY
C                             LEVEL RELATIVE TO INDEX LEVEL 1 (Rydbergs)
C           (R*8)  WUPPER   = SELECTED TRANSITION - UPPER ENERGY LEVEL
C                             STATISTICAL WEIGHT.
C           (R*8)  WLOWER   = SELECTED TRANSITION - LOWER ENERGY LEVEL
C                             STATISTICAL WEIGHT.
C           (R*8)  SUPPER   = 1/(UPPER LEVEL STATISTICAL WEIGHT)
```

```
C           (R*8)  SLOWER  = 1/(LOWER LEVEL STATISTICAL WEIGHT)
C           (R*8)  RYDDIF  = NEGATIVE TRANSITION ENERGY IN RYDBERGS
C                            ( NOTE: 1 Rydberg = 1.09737E5 cm-1)
C
C           (R*8)  ATE()   = EQUATION PARAMETER: DIMENSION - TEMPERATURE
C           (R*8)  GVAL()  = EQUATION PARAMETER (FOR UNIT GAMMA VALUE)
C                            DIMENSION - TEMPERATURE
C
C ROUTINES:  NONE
C
C NOTES:
C            EQUATIONS USED -
C
C                         2.17161E-8 x GAMMA x SQRT(157890 / TEMP)
C            RATE1 = ---------------------------------------------
C                    WLOWER x EXP( (EUPPER-ELOWER) * (157890 / TEMP) )
C
C                         2.17161E-8 x GAMMA x SQRT(157890 / TEMP)
C            DRATE1 =  ---------------------------------------
C                                      WUPPER
C
C            NOTE: OUTPUT VALUES ARE FOR 'GAMMA=1'
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 4569
C
C DATE:    18/09/90
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             I1A(NDTRN),  I2A(NDTRN),  ICNT,        MAXT
      INTEGER             NDLEV,       NDTEM,       NDTRN
      REAL*8              DRATE1(NDTEM,NDTRN),      ER(NDLEV)
      REAL*8              RATE1(NDTEM,NDTRN),       TEMP(NDTEM)
      REAL*8              XJA(NDLEV)
```

9.174 xxreia: Subroutine xxreia from library adaslib

```
      SUBROUTINE XXREIA( IUNIT , LSWIT , EIA )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXREIA *********************
C
C  PURPOSE: TO READ IN IONISATION POTENTIALS FROM AN INPUT MASTER
C           CONDENSED FILE ALLOCATED TO UNIT 'IUNIT' WITHOUT KNOWING HOW
C           MANY ARE PRESENT.
C
C  CALLING PROGRAMS: XXIN17 , XXIN80 , XXINST & GENERAL USE.
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  IUNIT   = UNIT TO WHICH CONDENSED FILE IS ALLOCATED
C  OUTPUT: (L*4)  LSWIT   = .TRUE.  => VALID SET OF 'EIA()' VALUES FOUND
C                           .FALSE. =>INVALID OR NO 'EIA()' VALUES FOUND
C  OUTPUT: (R*8)  EIA()   = IONISATION RATE COEFFICIENTS: ()=ION CHARGE
C                           UNITS - WAVE NUMBERS (CM-1)
C
C          (I*4)  IPOT    = INDEX OF IONISATION RATE COEFFT. VALUE BEING
C                           ANALYSED.
C          (I*4)  IREC    = NUMBER RECORDS READ IN FROM 'IUNIT'
C          (I*4)  IBGN    = FIRST BYTE IN 'STRING' CONTAINING THE 'EIA()
C                           VALUE BEING ANALYSED.
C          (I*4)  IEND    = LAST BYTE IN 'STRING' CONTAINING THE 'EIA()'
C                           VALUE BEING ANALYSED.
C                           (NOTE: 'EIA()' VALUES IF PRESENT ARE STORED
C                                  SIX TO A LINE AS 6F12.1)
C          (I*4)  IABT    = RETURN CODE FROM 'R8FCTN' FUNCTION.
C                           0 => NO ERROR
C                           2 => INVALID FLOATING POINT NUMBER
C                           (A BLANK INPUT STRING TO 'R8FCTN' CAUSES
C                            A ZERO VALUE TO BE RETURNED AND IABT=0)
C          (I*4)  I       = GENERAL ARRAY USE
C
C          (L*4)  LDEND   = .TRUE. =>VALID END TO 'EIA()' SECTION FOUND
C                           .FALSE.=>INVALID END TO 'EIA()'SECTION FOUND
C
C          (C*12) C12     = 12 BYTE STRING FOR STORING 'EIA()' VALUES
C          (C*80) STRING  = 80 BYTE STRING FOR STORING INPUT FILE RECORD
C
C
C NOTE:
C           ON RETURN THE NEXT RECORD SHOULD BE THAT CONTAINING THE
C           REDUCED DENSITY VALUES.
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          R8FCTN     ADAS      FUNCTION - CONVERT STRING -> REAL*8 NUM
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 4569
C
C DATE:    22/08/90
C
C UNIX-IDL PORT:
C
```

```
C VERSION: 1.1                          DATE: 06-09-95
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST RELEASE (NO CHANGES)
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      INTEGER             IUNIT
      LOGICAL             LSWIT
      REAL*8              EIA(50)
```

9.175 xxrepl: Subroutine xxrepl from library adaslib

```
      subroutine xxrepl( cstrg1 , csub1, csub2, cstrg2 , nsubs )

c-----------------------------------------------------------------------
c
c  ****************** fortran77 subroutine: xxrepl *********************
c
c  purpose: to replace all occurrences of a substring in a string with
c           an alternative substring
c
c  calling program: general use
c
c  subroutine:
c
c  input : (c*(*)) cstrg1   = input character string for conversion
c  input : (c*(*)) csub1    = character sub-string to replace
c  input : (c*(*)) csub2    = character sub-string to substitute
c
c  output: (c*(*)) cstrg2   = output string after conversion
c  output: (i*4)   nsubs    = number of substitutions made
c
c          (i*4)   i        = general use
c          (i*4)   ilen1    = length of string in bytes
c          (i*4)   ilen2    = length of string in bytes
c          (i*4)   nsubs_max= maximum number of subsitutions allowed
c
c routines:
c          routine    source    brief description
c          ------------------------------------------------------------
c          i4unit     adas      fetch unit number for output of messages
c
c note:
c
c
c author:  Hugh Summers,  University of Strathclyde
c          JA7.08
c          Tel. 0141-548-4196
c
c date  :  17/09/05
c
c version: 1.1                          date: 17/08/2005
c modified: Hugh Summers
c               - first edition.
c
c version: 1.2                          date: 03/01/2007
c modified: Hugh Summers
c               - remove redundant format statements.
c
c-----------------------------------------------------------------------
c-----------------------------------------------------------------------
c
c-----------------------------------------------------------------------
      CHARACTER*(*)       CSTRG1,      CSTRG2,      CSUB1,       CSUB2
      INTEGER             NSUBS
```

9.176 xxrmve: Subroutine xxrmve from library adaslib

```
      subroutine xxrmve( cstrg1 , cstrg2 , crmve )
C-----------------------------------------------------------------------
C
C  ****************** fortran77 subroutine: xxrmve *********************
C
C  purpose: to remove all occurrences of a selected character from a
C           string and concatenate.  Output string tail is blank filled
C
C  calling program: general use
C
C  subroutine:
C
C  input : (c*(*)) cstrg1   = input string for conversion
C  input : (c*1)   crmve    = character to be removed
C
C  output: (c*(*)) cstrg2   = output string after conversion
C
C          (i*4)   i        = general use
C          (i*4)   ilen     = length of 'cstrng' string in bytes
C
C routines:
C          routine    source    brief description
C          ------------------------------------------------------------
C          i4unit     adas      fetch unit number for output of messages
C
C
C
C author:  H. P. Summers, university of strathclyde
C          ja7.08
C          tel. 0141-548-4196
C
C date:    06/09/01
C
C version : 1.1
C date    : 06/09/2001
C modified: Hugh Summers
C               - first edition.
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      CHARACTER           CRMVE
      CHARACTER*(*)       CSTRG1,      CSTRG2
```

9.177 xxrmws: Subroutine xxrmws from library adaslib

```
C
      SUBROUTINE XXRMWS( STRING, CSTRING, LENGTH )

C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXRMWS *********************
C
C  PURPOSE: REMOVES ALL BLANKS IN INPUT STRING
C
C  CALLING PROGRAM: GENERAL USE.
C
C  INPUT    : (C*(*)) STRING = STRING.
C
C  OUTPUT   : (C*(*)) CSTRING = COMPRESSED STRING.
C
C  ROUTINES : NONE
C
C  AUTHOR   : Martin O'Mullane,
C             K1/1/43,
C             JET
C
C  VERSION  : 1.1                          DATE: 17/03/1999
C  MODIFIED : Martin O'Mullane
C             First version.
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      CHARACTER*(*)       CSTRING,     STRING
      INTEGER             LENGTH
```

9.178 xxrptn: Subroutine xxrptn from library adaslib

```
      subroutine  xxrptn( iunit  , ndstack,
     &                    ndptnl , ndptn  , ndptnc ,
     &                    nptnl  , nptn   , nptnc  ,
     &                    iptnla , iptna  , iptnca ,
     &                    lresol , lptn   ,
     &                    cstrg  ,
     &                    ncptn_stack     , cptn_stack
     &                  )
c-----------------------------------------------------------------------
c
c  ****************** fortran77 subroutine: xxprtn ********************
c
c  Purpose:  To read and analyse a partition block in a datafile header
c
c  Calling program: adas416
c
c  Notes:  (1) Partition levels, partitions and partition components are
c              labelled starting at 0 (but see (2)).
c          (2) Partition level 0 labels the resolved root partition level
c              partition level 1 labels the unresolved root partition
c              level.
c          (3) For an unresolved (standard) file, the partitions are each
c              ionisation stage from the neutral to the bare nucleus and
c              they are labelled by the ion charge.  Each partition has
c              just the one component.
c          (4) Distinguish the indexing (starting at 1) from the label
c              (starting at 0) .
c
c  Subroutine:
c
c  input : (i*4)  iunit     = unit to which input file is allocated
c  input : (i*4)  ndstack   = maximum no. of text lines in partition block
c
c  input : (i*4)  ndptnl    = maximum level of partitions
c  input : (i*4)  ndptn     = maximum no. of partitions in one level
c  input : (i*4)  ndptnc    = maximum no. of components in a partition
c  input : (l*4)  lresol    = .true.  => resolved root partition
c                           = .false. => standard root partition
c
c  output: (i*4)  nptnl     = number of partition levels in block
c  output: (i*4)  nptn()    = number of partitions in partition level
c                             1st dim: partition level
c  output: (i*4)  nptnc(,)  = number of components in partition
c                             1st dim: partition level
c                             2nd dim: member partition in partition level
c  output: (i*4)  iptnla()  = partition level label (0=resolved root,1=
c                                                    unresolved root)
c                             1st dim: partition level index
c  output: (i*4)  iptna(,)  = partition member label (labelling starts at 0)
c                             1st dim: partition level index
c                             2nd dim: member partition index in partition
c                             level
c  output: (i*4)  iptnca(,,)= component label (labelling starts at 0)
c                             1st dim: partition level index
c                             2nd dim: member partition index in partition
c                             level
c                             3rd dim: component index of member partition
c  output: (l*4)  lptn      = .true.  => partition block present
c                           = .false. => partition block not present
```

```
c  output: (c*80) cstrg     = string marking end of partition block
c  output: (i*4)  ncptn_stack= number of text lines in partition block
c  output: (c*80) cptn_stack()=text lines of partition block
c                              1st dim: text line pointer
c
c
c Routines:
c      Routine    Source Brief description
c      ------------------------------------------------------------
c      I4UNIT     ADAS Fetch unit number for output of messages
c      XXSLEN     ADAS Find non-blank characters in string
c      XXWORD     ADAS Extract position of number in buffer
c
c Author:  H. P. Summers, university of strathclyde
c          JA7.08
c          tel. 0141-548-4196
c
c Date:    25/08/05
c
c Version: 1.1   Date: 25/08/2005
c Modified: Hugh Summers
c   - First edition.
c
c Version: 1.2   Date: 28/02/2008
c Modified: Adam Foster
c   - Increased length of strg to 1024
c
c Version: 1.3   Date: 28/02/2008
c Modified: Allan Whiteford
c - Added comments for Adam's change
c               - Fixed capitalisation of comments section.
c
c-----------------------------------------------------------------------
c-----------------------------------------------------------------------
      CHARACTER*80        CPTN_STACK(NDSTACK),      CSTRG
      INTEGER             IPTNA(NDPTNL,NDPTN)
      INTEGER             IPTNCA(NDPTNL,NDPTN,NDPTNC)
      INTEGER             IPTNLA(NDPTNL),           IUNIT
      INTEGER             NCPTN_STACK, NDPTN,       NDPTNC,      NDPTNL
      INTEGER             NDSTACK,     NPTN(NDPTNL)
      INTEGER             NPTNC(NDPTNL,NDPTN),      NPTNL
      LOGICAL             LPTN,        LRESOL
```

9.179 xxsim: Subroutine xxsim from library adaslib

```
      SUBROUTINE XXSIM(A,IA,M,B,N,X,WKS,ERR,ISTOP)
C-----------------------------------------------------------------------
C
C ROUTINE: XXSIM
C
C PURPOSE: SOLVES THE SYSTEM OF SIMULTANEOUS EQUATIONS AX=B USING THE
C    NETLIB LINALG ROUTINE LSQR WHICH FOLLOWS THIS ROUTINE ALONG
C          WITH ITS DEPENDENCIES. THIS ROUTINE REPLACES NAG LIBRARY
C          ROUTINE F04ATF. HOWEVER, THE LU DECOMPOSITION IS NOT OUTPUT.
C
C CALLING PROGRAM: GENERAL USE
C
C INPUT:
C
C     (R*8)   A(,)    THE MATRIX A
C     (I*4)   IA      THE FIRST DIMENSION OF THE TWO-DIMENSIONAL
C                     ARRAY A, IA<=M
C     (I*4)   M       NUMBER OF ROWS OF A
C     (R*8)   B()     RIGHT-HAND-SIDE VECTOR B, DIMENSION = M
C     (I*4)   N       NUMBER OF COLUMNS OF A
C     (R*8)   WKS()   WORKSPACE VECTOR, DIMENSION >= M*N+2*N
C
C OUTPUT:
C
C     (R*8)   X()     SOLUTION VECTOR X, DIMENSION = N
C     (R*8)   ERR()   VECTOR OF THE ERROR ESTIMATES OF THE COMPONENTS OF
C                     X. SEE VARIABLE 'SE' IN LSQR.
C     (I*4)   ISTOP   AN ERROR CODE SET TO 0 OR 4 IF NO ERROR. SEE LSQR
C                     FOR A DESCRIPTION OF THE ERROR CODES.
C
C CALLS:
C     (EXT.)  MA      SUBROUTINE TO CALCULATE PRODUCTS OF A WITH GIVEN
C                     VECTORS. GIVEN NEXT IN THE FILE.
C ROUTINES:
C-----------------------------------------------------------------------
C     NAME     SOURCE   PURPOSE
C-----------------------------------------------------------------------
C     LSQR     NETLIB   CALCULATES THE SOLUTION - SEE BELOW
C-----------------------------------------------------------------------
C
C AUTHOR: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC.)
C
C DATE:   10-06-96
C
C VERSION 1.1                           DATE: 10-06-96
C MODIFIED: WILLIAM OSBORN
C           - FIRST VERSION
C VERSION 1.2                           DATE: 13-09-96
C MODIFIED: WILLIAM OSBORN
C           - ADDED IA PARAMETER SO THAT NON-SQUARE MATRICES CAN BE USED
C
C VERSION 1.3                           DATE: 25-09-96
C MODIFIED: WILLIAM OSBORN
C           - CHANGED DIMENSION OF B TO M
C
C VERSION 1.4                           DATE: 14-06-2000
C MODIFIED: Martin O'Mullane
C           - Changed iwk from real*8 to integer.
C
```

```
C-----------------------------------------------------------------------

      INTEGER             IA,           ISTOP,        M,            N
      REAL*8              A(IA,N),      B(M),         ERR(N)
      REAL*8              WKS(M*N+2*N),               X(N)
      INTEGER             IW(LENIW),    LENIW,        LENRW,        M
      INTEGER             MODE,         N
      REAL*8              RW(LENRW),    X(N),         Y(M)
      DOUBLE PRECISION    ACOND,        ANORM,        ARNORM,       ATOL
      DOUBLE PRECISION    BTOL,         CONLIM,       DAMP,         RNORM
      DOUBLE PRECISION    RW(LENRW),    SE(N),        U(M),         V(N)
      DOUBLE PRECISION    W(N),         X(N),         XNORM
      INTEGER             ISTOP,        ITN,          ITNLIM
      INTEGER             IW(LENIW),    LENIW,        LENRW,        M
      INTEGER             N,            NOUT
      DOUBLE PRECISION    X(N),         Y(N)
      INTEGER             INCX,         INCY,         N
      DOUBLE PRECISION    X(N)
      INTEGER             INCX,         N
      DOUBLE PRECISION    A,            X(N)
      INTEGER             INCX,         N
```

9.180 xxsion: Subroutine xxsion from library adaslib

```
      SUBROUTINE XXSION( SYMB , IZ , SION , LEN )
C
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXSION *********************
C
C  PURPOSE: RETURNS ION ELEMENT SYMBOL AND ION CHARGE AS A STRING
C           CONSTRUCTED AS FOLLOWS <SYMBOL><CHARGE>. IT ALSO RETURNS
C           THE LENGTH OF THE STRING.
C
C  CALLING PROGRAM: GENERAL USE
C
C  SUBROUTINE:
C
C  INPUT : (C*2)  SYMB      = ION ELEMENT SYMBOL OF BEAM.
C  INPUT : (I*4)  IZ        = ION CHARGE
C
C  INPUT : (C*4)  SION      = ION STRING '<SYMBOL><CHARGE>'.
C  INPUT : (I*4)  LEN       = LENGTH OF ION STRING.
C
C           (I*4)  I         = LENGTH IN BYTES OF NON-BLANK 'DSNAME'.
C
C ROUTINES: NONE
C
C AUTHOR:  JONATHAN NASH  (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/87
C          JET EXT. 5183
C
C DATE:    21/12/93
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 16-11-95
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST VERSION
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      CHARACTER*4         SION
      CHARACTER*2         SYMB
      INTEGER             IZ,          LEN
```

9.181 xxslen: Subroutine xxslen from library adaslib

```
      SUBROUTINE XXSLEN( CSTRNG , IFIRST , ILAST )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXSLEN *********************
C
C  PURPOSE: TO IDENTIFY THE FIRST AND LAST NON-BLANK CHARACTER IN A
C           STRING. (IF INPUT STRING IS BLANK IFIRST=ILAST=0)
C
C  CALLING PROGRAM: GENERAL USE
C
C  SUBROUTINE:
C
C  INPUT : (C*(*)) CSTRNG   = INPUT STRING FOR INTERROGATION
C
C  OUTPUT: (I*4)   IFIRST   = BYTE POSITION OF FIRST NON-BLANK CHARACTER
C                             IN INPUT STRING.
C  OUTPUT: (I*4)   ILAST    = BYTE POSITION OF LAST  NON-BLANK CHARACTER
C                             IN INPUT STRING.
C
C          (I*4)   I        = GENERAL USE
C          (I*4)   ILEN     = LENGTH OF 'CSTRNG' STRING IN BYTES
C
C ROUTINES: NONE
C
C NOTE:
C
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/37
C          JET EXT. 6023
C
C DATE  :  06/07/93
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      CHARACTER*(*)       CSTRNG
      INTEGER             IFIRST,      ILAST
```

9.182 xxsort: Subroutine xxsort from library adaslib

```
      subroutine xxsort(n,arr,brr)

C-----------------------------------------------------------------------
C
C    ***************** fortran77 subroutine: xxsort *****************
C
C  purpose: Sorts array 'arr' into ascending numerical order, with
C           corresponding rearrangement of 'brr'. Uses Shell's method.
C
C  calling program: various
C
C  input        : (i*4) n      = number of array elements
C
C  input/output : (r*8) arr()  = array to be sorted into ascending order
C  input/output : (r*8) brr()  = array corresponding to 'arr'
C
C
C  author: Paul Bryans, University of Strathclyde
C
C  date:   02/12/04
C
C  update:
C
C-----------------------------------------------------------------------
      INTEGER             N
      REAL*8              ARR(N),      BRR(N)
```

9.183 xxspec: Subroutine xxspec from library adaslib

```
      SUBROUTINE XXSPEC( USRGRP , USRTYP , USREXT )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXSPEC *********************
C
C  PURPOSE: ADAS ROUTINE - SETS UP THE DEFAULT USEGRP, USRTYP AND USREXT
C           WHICH IDENTIFY THE FILENAME AND EXTENSION TO BE READ IN
C           SUBROUTINE SPEC.  IT WORKS IN THE SAME MANNER AS XXUID WHICH
C           WHICH ALLOWS THE DEFAULT USER SPACE TO BE SET
C
C
C           USRGRP: VALUE ON INPUT  =>  USRGRP: VALUE ON OUTPUT
C
C                     ?                 CURRENT ADAS DATA GROUPNAME
C                     *                 DEFAULT ADAS DATA GROUPNAME
C                  <BLANK>              *** USRGRP VALUE NOT CHANGED ***
C                  <OTHER>              *** USRGRP VALUE NOT CHANGED ***
C
C
C           USRTYP: VALUE ON INPUT  =>  USRTYP: VALUE ON OUTPUT
C
C                     ?                 CURRENT ADAS DATA TYPENAME
C                     *                 DEFAULT ADAS DATA TYPENAME
C                  <BLANK>              *** USRTYP VALUE NOT CHANGED ***
C                  <OTHER>              *** USRTYP VALUE NOT CHANGED ***
C
C
C           USREXT: VALUE ON INPUT  =>  USREXT: VALUE ON OUTPUT
C
C                     ?                 CURRENT ADAS DATA MEMBER EXTENS.
C                     *                 DEFAULT ADAS DATA MEMBER EXTENS
C                  <BLANK>              *** USREXT VALUE NOT CHANGED ***
C                  <OTHER>              *** USREXT VALUE NOT CHANGED ***
C
C         ? => QUERIES CURRENT ADAS DATA USRGRP, USRTYP OR USREXT
C              SETTING.
C         * => SETS ADAS DATA USEGRP, USRTYP OR USREXT SETTING
C              TO DEFAULT VALUE.
C   <BLANK> => SETS ADAS DATA USRGRP, USRTYP OR USREXT SETTING
C              TO DEFAULT VALUE.
C   <OTHER> => SETS ADAS DATA USRGRP, USRTYP OR USREXT SETTING
C              TO INPUT   VALUE.
C
C  CALLING PROGRAM: SPEC AND MAIN PROGRAMS USING SPEC, ADAS503
C
C  SUBROUTINE:
C
C  I/O   : (C*8)  USRGRP   = USRFIL UNDER WHICH ADAS DATA IS STORED
C                            (IF BLANK DEFAULTS TO DEFGRP)
C
CA I/O   : (C*80) USRTYP   = SUBDIRECTORY (OPTIONAL) WHERE ADAS DATA
CA              FILE IS LOCATED. (IF BLANK DEFAULTS TO
CA      DEFTYP)
C
C  I/O   : (C*3)  USREXT   = USREXT UNDER WHICH ADAS DATA IS STORED
C                            (IF BLANK DEFAULTS TO DEFEXT)
C
C          (C*8)  DEFGRP   = PARAMETER = DEFAULT USER GROUP FOR ADAS
C                                        DATA SOURCE
```

```
C
C          (C*80) DEFTYP   = PARAMETER = DEFAULT SUBDIRECTORY OF ADAS
C                                        DATA SOURCE
C
C          (C*3)  DEFEXT   = PARAMETER = DEFAULT USER EXTENSION FOR ADAS
C                                        DATA SOURCE
C
C          (C*8)  ADASGR   = CURRENT ADAS DATA SOURCE GROUP
CA         (C*80) ADASTY   = CURRENT ADAS DATA SOURCE TYPE
C          (C*3)  ADASEX   = CURRENT ADAS DATA SOURCE EXTENSION
C
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C
C NOTE:
C          TO CHECK CURRENT ADAS SOURCE USRGRP, USRTYP AND USREXT
C          CALL XXSPEC WITH ?'S AS INPUTS.
C
C AUTHOR:  HUGH P. SUMMERS, JET
C          K1/1/57
C          JET EXT. 4941
C
C DATE:     2/09/93
C
C UPDATE:  L. JALOTA - 1/11/94 (TESSELLA SUPPORT SERVICES PLC)
C                      CHANGED VALUES OF DEFGRP,DEFTYP, DEFEXT SUITABLE
C                      FOR DEC ALPHA DIRECTORY STRUCTURE.
C
C UPDATE:  L.JALOTA - 23/11/94 : TIDIED UP STRING LENGTH DEFINITIONS.
C-----------------------------------------------------------------------
      CHARACTER*3         USREXT
      CHARACTER*8         USRGRP
      CHARACTER*80        USRTYP
```

9.184 xxsple: Subroutine xxsple from library adaslib

```
      SUBROUTINE XXSPLE( LSETX , IOPT   , FINTX ,
     &                   NIN   , XIN    , YIN   ,
     &                   NOUT  , XOUT   , YOUT  ,
     &                   DY    , LINTRP
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXSPLE *********************
C
C  PURPOSE:           TO INTERPOLATE/EXTRAPOLATE USING CUBIC SPLINES
C
C                     (IF IOPT < 0 NO EXTRAPOLATION TAKES PLACE = VALUES
C                      SET TO ZERO).- LOGICAL ARRAY 'LINTRP()' SPECIFIES
C                      WHETHER OUTPUT SPLINE IS INTERPOLATED '.TRUE.' OR
C                      EXTRAPOLATED '.FALSE.'.
C
C                      (AS FOR 'XXSPLN' EXCEPT 'LINTRP' ARGUMENT ADDED).
C
C  CALLING PROGRAMS:  GENERAL USE
C
C  SUBROUTINE:
C
C  I/O   : (L*4)  LSETX   = .TRUE.  => SET UP SPLINE PARAMETERS RELATING
C                                      TO 'XIN' AXIS.
C                           .FALSE. => DO NOT SET UP SPLINE PARAMETERS
C                                      RELATING TO 'XIN' AXIS.
C                                      (I.E. THEY WERE SET IN A PREVIOUS
C                                            CALL )
C                           ( 'LSETX' IS ALWAYS RETURN AS '.FALSE.'  ON
C                             RETURN FROM THE SUBROUTINE ).
C                           ** IMPORTANT: SEE NOTES BELOW ON 'LSETX' **
C  INPUT : (I*4)  IOPT    = SPLINE END CONDITIONS/EXTRAPOLATION CONTROL
C                           SWITCH - SEE NOTES BELOW
C                           I.E. DEFINES THE BOUNDARY DERIVATIVES.
C                                (VALID VALUES = 0, 1, 2, 3, 4)
C                           IF IOPT < 0 THEN NO EXTRAPOLATION TAKES
C                           - ANY VALUES REQUIRING EXTRAPOLATION WILL BE
C                             SET TO ZERO (END CONDITIONS AS FOR IOPT=0)
C  INPUT : (R*8)  FINTX   = INTERPOLATING X-COORDINATE TRANSFORMATION.
C                           EXTERNAL FUNCTION (SEE ROUTINES BELOW)
C
C  INPUT : (I*4)  NIN     = NUMBER OF KNOTS
C  INPUT : (R*8)  XIN()   = X-VALUES OF KNOTS
C  INPUT : (R*8)  YIN()   = Y-VALUES OF KNOTS
C
C  INPUT : (I*4)  NOUT    = NUMBER OF OUTPUT VALUES TO BE INTERPOLATED
C                           EXTRAPOLATED.
C  INPUT : (R*8)  XOUT()  = X-VALUES AT WHICH INTERPOLATION/EXTRAPOLA-
C                           TION REQUIRED
C  OUTPUT: (R*8)  YOUT()  = INTERPOLATED/EXTRAPOLATED Y-VALUES FOR
C                           REQUESTED 'XOUT()' VALUES.
C
C  OUTPUT: (R*8)  DY()    = DERIVATIVES AT INPUT KNOTS (ARRAY SIZE: NIN)
C  OUTPUT: (L*4)  LINTRP()= .TRUE.  => 'YOUT()' VALUE INTERPOLATED.
C                           .FALSE. => 'YOUT()' VALUE EXTRAPOLATED.
C                           (ARRAY SIZE: NOUT)
C
C          (I*4)  NKNOTS  = PARAMETER = MAXIMUM  NUMBER OF KNOTS ALLOWED
C          (I*4)  NIOPT   = PARAMETER = MAXIMUM  VALUE OF IOPT ALLOWED
```

```
C
C          (I*4)  I       = GENERAL ARRAY USE
C          (I*4)  K       = INDEX OF 'XOUT()' VALUE FOR INTERPOLATION/
C                           EXTRAPOLATION.
C          (I*4)  NIN0    = 'NIN' - 1
C          (I*4)  INTER   = INDEX OF CLOSEST/NEXT HIGHEST VALUE OF
C                           'XIN()' TO THE VALUE OF 'XOUT()' BEING
C                           INTERPOLATED/EXTRAPOLATED.
C          (I*4)  NOPT    = VALUE OF  'IOPT' USED IN  CALCULATING  END-
C                           CONDITIONS   FOR  STORED  'X-VALUE'  SPLINE
C                           PARAMETERS.   (NOTE:  IF  'IOPT < 0',  THEN
C                           'NOPT = 0'.) - I.E. 'NOPT = MAX( 0, IOPT )'.
C
C          (R*8)  XK      = VALUE OF 'XOUT(K)' BEING INTERPOLATED/
C                           EXTRAPOLATED
C          (R*8)  XKK     = TRANSFORMED VALUE OF 'XOUT(K)' BEING
C                           INTERPOLATED/EXTRAPOLATED.
C          (R*8)  T1      = INVERSE OF SEPARATION OF KNOTS EITHER
C                           SIDE OF CURRENT KNOT.
C          (R*8)  T2      = (CURRENT KNOT POSITION TO NEXT HIGHEST KNOT
C                            POSITION) DIVIDED BY 'T1'
C          (R*8)  T3      = (CURRENT KNOT POSITION TO NEXT LOWEST  KNOT
C                            POSITION) DIVIDED BY 'T1'
C          (R*8)  T4      = INTERPOLATION FACTOR FOR CURRENT KNOT
C          (R*8)  DL1     = (REQUESTED 'XOUT()' VALUE TO NEXT HIGHEST
C                            KNOT POSITION) DIVIDED BY SEPERATION OF
C                            KNOTS EITHER SIDE OF 'XOUT(K)'.
C          (R*8)  DL2     = (REQUESTED 'XOUT()' VALUE TO NEXT LOWEST
C                            KNOT POSITION) DIVIDED BY SEPERATION OF
C                            KNOTS EITHER SIDE OF 'XOUT(K)'.
C          (R*8)  DL2     = (REQUESTED 'XOUT()' VALUE TO NEXT LOWEST
C          (R*8)  DL3     =  SEPERATION OF KNOTS EITHER SIDE OF
C                           'XOUT(K)' * 'DL1' * 'DL2'.
C
C          (L*4)  LEXTRP  = .TRUE.  => 'EXTRAPOLATION SWITCHED ON'.
C                           .FALSE. => 'EXTRAPOLATION SWITCHED OFF'.
C
C          (R*8)  QVAL()  = VALUE OF 'Q(1)'   : FUNCTION OF 'NOPT'
C          (R*8)  D2VAL() = VALUE OF 'D2(1)'  : FUNCTION OF 'NOPT'
C          (R*8)  D3VAL() = VALUE OF 'D3(1)'  : FUNCTION OF 'NOPT'
C          (R*8)  UVAL() =  VALUE OF 'U(NIN)' : FUNCTION OF 'NOPT'
C          (R*8)  AGRL() =  POLYNOMIAL CONSTANTS FOR CUBIC SPLINE FOR
C                           GIVEN 'XOUT(K)' VALUE.
C          (R*8)  X()     =  TRANSFORMED VALUES OF 'XIN()'
C          (R*8)  H()     =  SEPERATION, ALONG X-AXIS, OF KNOT FROM NEXT
C                           HIGHEST KNOT.
C          (R*8)  Q()     =  SECOND DERIVATIVE FOR KNOT
C          (R*8)  U()     =  TEMPORARY STORAGE OF DECOMPOSED FACTORS
C          (R*8)  DELY() =  SEPERATION, ALONG Y-AXIS, OF KNOT FROM NEXT
C                           HIGHEST KNOT.
C          (R*8)  D1()    =  MULTIPLICATION FACTOR USED IN CALCULATING
C                           'U()'.
C          (R*8)  D2()    =  MULTIPLICATION FACTOR USED IN CALCULATING
C                           'U()'.
C          (R*8)  D3()    =  MULTIPLICATION FACTOR USED IN CALCULATING
C                           'U()'.
C
C          (L*4)  LUVAL()=  .TRUE. => VALUE OF 'UVAL()' REFERS TO RATE
C                                     OF CHANGE OF SLOPE AT FINAL POINT.
C                           .FALSE.=> VALUE OF 'UVAL()' REFERS TO FINAL
C                                     SLOPE
```

```
C                          FUNCTION OF 'NOPT'
C
C NOTES: 'LSETX': SET TO .TRUE. ON ENTRY IF A NEW 'XIN' ARRAY IS BEING
C                 USED.  IF THE 'XIN' AXIS IS THE SAME FOR A NUMBER OF
C                 CALLS THEN DO NOT RESET 'LSETX'  -  THIS  SUBROUTINE
C                 SETS IT TO .FALSE. FOR YOU.   IF THE VALUE OF 'NOPT'
C                 IS CHANGED BETWEEN CALLS THEN THE VALUE  OF  'LSETX'
C                 ON  ENTRY IS TAKEN AS BEING EQUAL TO .TRUE. .
C
C                 THEREFORE 'LSETX' NEED ONLY BE SET TO .TRUE. ON ENTRY
C                 IF EITHER IT IS ITS FIRST CALL OR IF ANY ONE  OF  THE
C                 FOLLOWING VALUES HAS CHANGED:
C
C                 'NIN' , 'FINTX' , 'XIN(I), I=1,NIN'
C
C                 CARE: A VARIABLE MUST BE USED FOR 'LSETX', A CONSTANT,
C                       I.E.  .TRUE. ,  CANNOT BE DIRECTLY TYPED AS  AN
C                       ARGUMENT BECAUSE IT WILL BE CHANGED TO  .FALSE.
C                       ON RETURN.
C
C          SPLINE  END CONDITIONS AND EXTRAPOLATION DEPEND ON 'IOPT' AS
C          FOLLOWS:
C
C          ------------------------------------------------------------
C          | IOPT  | NOPT |  DY(1)  DDY(1)  |  DY(N)   DDY(N)  |EXTRAP'N|
C          |-------|------|-----------------|------------------|--------|
C          | < 0   |   0  |    -     0.0    |    -      0.0    |  NO    |
C          |   0   |   0  |    -     0.0    |    -      0.0    |  YES   |
C          |   1   |   1  |    -     0.0    |  -1.5     -      |  YES   |
C          |   2   |   2  |   0.0    -      |   1.0     -      |  YES   |
C          |   3   |   3  |  -0.5    -      |  -1.5     -      |  YES   |
C          |   4   |   4  |   0.0    -      |    -      0.0    |  YES   |
C          |   5   |   5  |  -4.5    -      |  -1.5     -      |  YES   |
C          |   6   |   6  |  +0.5    -      |    -      0.0    |  YES   |
C          |   7   |   7  |  -3.5    -      |    -      0.0    |  YES   |
C          ------------------------------------------------------------
C
C             NB. OPTIONS TO BE EXTENDED FOR POWER AND CX APPLICATION
C
C          ------------------------------------------------------------
C           IF ( IOPT.LT.0 ) - NO EXTRAPOLATION TAKES PLACE VALUES SET
C                              TO ZERO (CARE IF LOG OF OUTPUT IS NEEDED).
C           IF ( IOPT.GT.7 ) PROGRAM STOPS
C          ------------------------------------------------------------
C
C           THIS SUBROUTINE IS AN AMENDED  AND STRUCTURED VERSION OF  THE
C           SUBROUTINE  'ESPLINE'  WRITTEN BY  H.P. SUMMERS,   JET   26TH
C           OCTOBER 1989.   IT REMOVES THE COMMON BLOCK  /IONSPL/ ,   THE
C           SWITCHES 'ISW & ISW2' AND ALSO THE CASE FOR THE INTERPOLATION
C           OF CHARGE STATE VALUES.   IT INTRODUCES THE FEATURE  THAT  AN
C           ARRAY OF INPUT  'X-VALUES'  CAN BE  INTERPOLATED/EXTRAPOLATED
C           IN ONE CALL.
C
C ROUTINES:
C           ROUTINE    SOURCE    BRIEF DESCRIPTION
C           ------------------------------------------------------------
C           FINTX      ------    EXTERNAL  REAL*8  FUNCTION,   USED  TO
C                                TRANSFORM X-COORDINATES.
C
C
C AUTHOR:   PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
```

```
C           K1/0/81
C           JET EXT. 4569
C
C DATE:     14/01/91 - ADAS91: AS FOR 'XXSPLN' BUT WITH 'LINTRP()' ADDED
C
C VERSION:  1.2
C
C MODIFIED: LORNE HORTON (JET)  DATE: 25/10/97
C           - ADDED IOPT CHOICES 5, 6 AND 7
C
C VERSION:  1.3
C
C MODIFIED: Martin O'Mullane (JET)  DATE: 2/6/99
C           - SAVE nin0 and inter variables also. All compilers, ie
C             especially g77, do not automatically save (or initialise
C             variables to zero).
C
C VERSION  : 1.4
C DATE     : 10-04-2007
C MODIFIED : Allan Whiteford
C               - Modified documentation as part of automated
C   subroutine documentation preparation.
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IOPT,        NIN,         NOUT
      LOGICAL             LINTRP(NOUT),             LSETX
      REAL*8              DY(NIN),     XIN(NIN),    XOUT(NOUT)
      REAL*8              YIN(NIN),    YOUT(NOUT)
```

9.185 xxsplf: Subroutine xxsplf from library adaslib

```
      SUBROUTINE XXSPLF( LSETX , LSETY  , IOPT   , FINTX ,
     &                   NIN   , XIN    , YIN    ,
     &                   NOUT  , XOUT   , YOUT   ,
     &                   X     , DY     ,
     &                   Q     , D1     , D2     , D3    ,
     &                   LINTRP
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXSPLF *********************
C
C  PURPOSE:           TO INTERPOLATE/EXTRAPOLATE USING CUBIC SPLINES
C
C                     (IF IOPT < 0 NO EXTRAPOLATION TAKES PLACE = VALUES
C                      SET TO ZERO).- LOGICAL ARRAY 'LINTRP()' SPECIFIES
C                      WHETHER OUTPUT SPLINE IS INTERPOLATED '.TRUE.' OR
C                      EXTRAPOLATED '.FALSE.'.
C
C                      (AS FOR 'XXSPLN' EXCEPT 'LINTRP' ARGUMENT ADDED).
C                      (AS FOR 'XXSPLE' EXCEPT WITH OPTION TO USE
C                       PREVIOUSLY CALCULATED SPLINE DERIVATIVES)
C
C  CALLING PROGRAMS:  GENERAL USE
C
C  SUBROUTINE:
C
C  I/O   : (L*4)  LSETX   = .TRUE.  => SET UP SPLINE PARAMETERS RELATING
C                                      TO 'XIN' AXIS.
C                           .FALSE. => DO NOT SET UP SPLINE PARAMETERS
C                                      RELATING TO 'XIN' AXIS.
C                                      (I.E. THEY WERE SET IN A PREVIOUS
C                                            CALL )
C                           ( 'LSETX' IS ALWAYS RETURN AS '.FALSE.'  ON
C                             RETURN FROM THE SUBROUTINE ).
C                           ** IMPORTANT: SEE NOTES BELOW ON 'LSETX' **
C  I/O   : (L*4)  LSETY   = .TRUE.  => CALCULATE SPLINE DERIVATIVES
C                                      RELATING TO 'YIN' AXIS.
C                           .FALSE. => DO NOT SET UP SPLINE DERIVATIVES
C                                      RELATING TO 'YIN' AXIS.
C                                      (I.E. THEY WERE SET IN A PREVIOUS
C                                            CALL )
C                           ( 'LSETY' IS ALWAYS RETURN AS '.FALSE.'  ON
C                             RETURN FROM THE SUBROUTINE ).
C                           ** IMPORTANT: SEE NOTES BELOW ON 'LSETY' **
C  INPUT : (I*4)  IOPT    = SPLINE END CONDITIONS/EXTRAPOLATION CONTROL
C                           SWITCH - SEE NOTES BELOW
C                           I.E. DEFINES THE BOUNDARY DERIVATIVES.
C                                (VALID VALUES = 0, 1, 2, 3, 4)
C                           IF IOPT < 0 THEN NO EXTRAPOLATION TAKES
C                           - ANY VALUES REQUIRING EXTRAPOLATION WILL BE
C                             SET TO ZERO (END CONDITIONS AS FOR IOPT=0)
C  INPUT : (R*8)  FINTX   = INTERPOLATING X-COORDINATE TRANSFORMATION.
C                           EXTERNAL FUNCTION (SEE ROUTINES BELOW)
C
C  INPUT : (I*4)  NIN     = NUMBER OF KNOTS
C  INPUT : (R*8)  XIN()   = X-VALUES OF KNOTS
C  INPUT : (R*8)  YIN()   = Y-VALUES OF KNOTS
C
C  INPUT : (I*4)  NOUT    = NUMBER OF OUTPUT VALUES TO BE INTERPOLATED
```

```
C                              EXTRAPOLATED.
C  INPUT : (R*8)  XOUT()  = X-VALUES AT WHICH INTERPOLATION/EXTRAPOLA-
C                              TION REQUIRED
C  OUTPUT: (R*8)  YOUT()  = INTERPOLATED/EXTRAPOLATED Y-VALUES FOR
C                              REQUESTED 'XOUT()' VALUES.
C
C  I/O   : (R*8)  X()     = TRANSFORMED VALUES OF 'XIN()'. (ARRAY SIZE:
C                              NIN) REQUIRED INPUT IF LSETX IS .FALSE.
C  I/O   : (R*8)  DY()    = DERIVATIVES AT INPUT KNOTS. REQUIRED INPUT
C                              IF LSETY IS .FALSE.
C  I/O   : (R*8)  Q()     = SECOND DERIVATIVE FOR KNOT. REQUIRED INPUT
C                              IF LSETX IS .FALSE. AND LSETY IS .TRUE.
C  I/O   : (R*8)  D1()    = MULTIPLICATION FACTOR USED IN CALCULATING
C                              'U()'. REQUIRED INPUT IF LSETX IS .FALSE.
C                              AND LSETY IS .TRUE.
C  I/O   : (R*8)  D2()    = MULTIPLICATION FACTOR USED IN CALCULATING
C                              'U()'. REQUIRED INPUT IF LSETX IS .FALSE.
C                              AND LSETY IS .TRUE.
C  I/O   : (R*8)  D3()    = MULTIPLICATION FACTOR USED IN CALCULATING
C                              'U()'. REQUIRED INPUT IF LSETX IS .FALSE.
C                              AND LSETY IS .TRUE.
C
C  OUTPUT: (L*4)  LINTRP()= .TRUE.  => 'YOUT()' VALUE INTERPOLATED.
C                              .FALSE. => 'YOUT()' VALUE EXTRAPOLATED.
C                              (ARRAY SIZE: NOUT)
C
C          (I*4)  NKNOTS  = PARAMETER = MAXIMUM  NUMBER OF KNOTS ALLOWED
C          (I*4)  NIOPT   = PARAMETER = MAXIMUM  VALUE OF IOPT ALLOWED
C
C          (I*4)  I       = GENERAL ARRAY USE
C          (I*4)  K       = INDEX OF 'XOUT()' VALUE FOR INTERPOLATION/
C                              EXTRAPOLATION.
C          (I*4)  NIN0    = 'NIN' - 1
C          (I*4)  INTER   = INDEX OF CLOSEST/NEXT HIGHEST VALUE OF
C                              'XIN()' TO THE VALUE OF 'XOUT()' BEING
C                              INTERPOLATED/EXTRAPOLATED. WHEN LOOPING
C                              OVER MULTIPLE YOUT EVALUATIONS, THE INDEX
C                              OF THE LAST EVALUATION IS USED AS THE
C                              INITIAL GUESS FOR THE NEXT.
C          (I*4)  NOPT    = VALUE OF  'IOPT' USED IN  CALCULATING  END-
C                              CONDITIONS   FOR  STORED  'X-VALUE'  SPLINE
C                              PARAMETERS.   (NOTE:  IF  'IOPT < 0',  THEN
C                              'NOPT = 0'.) - I.E. 'NOPT = MAX( 0, IOPT )'.
C
C          (R*8)  XK      = VALUE OF 'XOUT(K)' BEING INTERPOLATED/
C                              EXTRAPOLATED
C          (R*8)  XKK     = TRANSFORMED VALUE OF 'XOUT(K)' BEING
C                              INTERPOLATED/EXTRAPOLATED.
C          (R*8)  T1      = INVERSE OF SEPARATION OF KNOTS EITHER
C                              SIDE OF CURRENT KNOT.
C          (R*8)  T2      = (CURRENT KNOT POSITION TO NEXT HIGHEST KNOT
C                             POSITION) DIVIDED BY 'T1'
C          (R*8)  T3      = (CURRENT KNOT POSITION TO NEXT LOWEST  KNOT
C                             POSITION) DIVIDED BY 'T1'
C          (R*8)  T4      = INTERPOLATION FACTOR FOR CURRENT KNOT
C          (R*8)  DL1     = (REQUESTED 'XOUT()' VALUE TO NEXT HIGHEST
C                             KNOT POSITION) DIVIDED BY SEPARATION OF
C                             KNOTS EITHER SIDE OF 'XOUT(K)'.
C          (R*8)  DL2     = (REQUESTED 'XOUT()' VALUE TO NEXT LOWEST
C                             KNOT POSITION) DIVIDED BY SEPARATION OF
C                             KNOTS EITHER SIDE OF 'XOUT(K)'.
```

```
C          (R*8)  DL2     = (REQUESTED 'XOUT()' VALUE TO NEXT LOWEST
C          (R*8)  DL3     =  SEPARATION OF KNOTS EITHER SIDE OF
C                            'XOUT(K)' * 'DL1' * 'DL2'.
C
C          (L*4)  LEXTRP  = .TRUE.  => 'EXTRAPOLATION SWITCHED ON'.
C                           .FALSE. => 'EXTRAPOLATION SWITCHED OFF'.
C
C          (R*8)  QVAL()  = VALUE OF 'Q(1)'   : FUNCTION OF 'NOPT'
C          (R*8)  D2VAL() = VALUE OF 'D2(1)'  : FUNCTION OF 'NOPT'
C          (R*8)  D3VAL() = VALUE OF 'D3(1)'  : FUNCTION OF 'NOPT'
C          (R*8)  UVAL() =  VALUE OF 'U(NIN)' : FUNCTION OF 'NOPT'
C          (R*8)  AGRL() =  POLYNOMIAL CONSTANTS FOR CUBIC SPLINE FOR
C                           GIVEN 'XOUT(K)' VALUE.
C          (R*8)  H()    =  SEPARATION, ALONG X-AXIS, OF KNOT FROM NEXT
C                           HIGHEST KNOT.
C          (R*8)  HINTER =  SEPARATION, ALONG X-AXIS, IN INTERVAL FOR
C                           INTERPOLATION
C          (R*8)  U()    =  TEMPORARY STORAGE OF DECOMPOSED FACTORS
C          (R*8)  DELY() =  SEPARATION, ALONG Y-AXIS, OF KNOT FROM NEXT
C                           HIGHEST KNOT.
C
C          (L*4)  LUVAL()=  .TRUE. => VALUE OF 'UVAL()' REFERS TO RATE
C                                     OF CHANGE OF SLOPE AT FINAL POINT.
C                           .FALSE.=> VALUE OF 'UVAL()' REFERS TO FINAL
C                                     SLOPE
C                            FUNCTION OF 'NOPT'
C
C NOTES: 'LSETX': SET TO .TRUE. ON ENTRY IF A NEW 'XIN' ARRAY IS BEING
C                 USED.  IF THE 'XIN' AXIS IS THE SAME FOR A NUMBER OF
C                 CALLS THEN DO NOT RESET 'LSETX'  -  THIS  SUBROUTINE
C                 SETS IT TO .FALSE. FOR YOU.   IF THE VALUE OF 'NOPT'
C                 IS CHANGED BETWEEN CALLS THEN THE VALUE  OF  'LSETX'
C                 ON  ENTRY IS TAKEN AS BEING EQUAL TO .TRUE.  NOPT IS
C                 INITIALISED TO -1  SO  THAT LSETX WILL BE SET .TRUE.
C                 ON THE FIRST CALL OF THIS SUBROUTINE.
C
C                 THEREFORE 'LSETX' NEED ONLY BE SET TO .TRUE. ON ENTRY
C                 IF ANY ONE OF THE FOLLOWING VALUES HAS CHANGED:
C
C                 'NIN' , 'FINTX' , 'XIN(I), I=1,NIN'
C
C        'LSETY': SET TO .TRUE. ON ENTRY IF A NEW 'YIN' ARRAY IS BEING
C                 USED.  IF THE 'YIN' AXIS IS THE SAME FOR A NUMBER OF
C                 CALLS THEN DO NOT RESET 'LSETY'  -  THIS  SUBROUTINE
C                 SETS IT  TO  .FALSE.  FOR YOU.   IF LSETX IS .TRUE.,
C                 EITHER  ON  ENTRY  OR  BECAUSE THE ROUTINE RESETS IT
C                 (SEE ABOVE) THEN LSETY IS ALSO SET TO .TRUE.
C
C                 THEREFORE 'LSETY' NEED ONLY BE SET TO .TRUE. ON ENTRY
C                 IF YIN HAS CHANGED WHILE THE 'X' VALUES HAVE NOT.
C
C                 CARE: VARIABLES MUST BE USED FOR 'LSETX' AND 'LSETY',
C                       A CONSTANT, I.E.  .TRUE. ,  CANNOT BE DIRECTLY
C                       TYPED AS AN ARGUMENT BECAUSE IT WILL BE CHANGED
C                       TO  .FALSE. ON RETURN.
C
C          SPLINE  END CONDITIONS AND EXTRAPOLATION DEPEND ON 'IOPT' AS
C          FOLLOWS:
C
C          ---------------------------------------------------------------
C          | IOPT  | NOPT |  DY(1)  DDY(1)  |  DY(N)   DDY(N)  |EXTRAP'N|
```

```
C         |-------|------|-----------------|------------------|--------|
C         | < 0   |   0  |    -     0.0    |    -      0.0    |  NO    |
C         |   0   |   0  |    -     0.0    |    -      0.0    |  YES   |
C         |   1   |   1  |    -     0.0    |  -1.5      -     |  YES   |
C         |   2   |   2  |   0.0     -     |   1.0      -     |  YES   |
C         |   3   |   3  |  -0.5     -     |  -1.5      -     |  YES   |
C         |   4   |   4  |   0.0     -     |    -      0.0    |  YES   |
C         |   5   |   5  |  -4.5     -     |  -1.5      -     |  YES   |
C         |   6   |   6  |  +0.5     -     |    -      0.0    |  YES   |
C         |   7   |   7  |  -3.5     -     |    -      0.0    |  YES   |
C         ------------------------------------------------------------
C
C            NB. OPTIONS TO BE EXTENDED FOR POWER AND CX APPLICATION
C
C         ------------------------------------------------------------
C          IF ( IOPT.LT.0 ) - NO EXTRAPOLATION TAKES PLACE VALUES SET
C                             TO ZERO (CARE IF LOG OF OUTPUT IS NEEDED).
C          IF ( IOPT.GT.7 ) PROGRAM STOPS
C         ------------------------------------------------------------
C
C          THIS SUBROUTINE IS AN AMENDED  AND STRUCTURED VERSION OF  THE
C          SUBROUTINE  'ESPLINE'  WRITTEN BY  H.P. SUMMERS,   JET   26TH
C          OCTOBER 1989.   IT REMOVES THE COMMON BLOCK  /IONSPL/ ,   THE
C          SWITCHES 'ISW & ISW2' AND ALSO THE CASE FOR THE INTERPOLATION
C          OF CHARGE STATE VALUES.   IT INTRODUCES THE FEATURE  THAT  AN
C          ARRAY OF INPUT  'X-VALUES'  CAN BE  INTERPOLATED/EXTRAPOLATED
C          IN ONE CALL.
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          FINTX      ------    EXTERNAL  REAL*8  FUNCTION,   USED  TO
C                               TRANSFORM X-COORDINATES.
C          XXHUNT     ------    SEARCH ROUTINE FOR FINDING INTERVAL
C                               CONTAINING A PRESCRIBED VALUE IN A
C                               MONOTONIC VECTOR.  INITIAL GUESSES ARE
C                               USED TO SPEED THE SEARCH.
C
C
C AUTHOR:   LORNE D. HORTON (IPP GARCHING)
C           L5.213
C           IPP EXT. 1635
C DATE:     18/03/03
C
C-----------------------------------------------------------------------
C Notes:  AS FOR 'XXSPLE' BUT WITH 'LSETY' ADDED TO ALLOW
C         EXTERNAL SAVING OF SPLINE COEFFICIENTS (FOR
C         EXAMPLE, WHEN IT IS NECESSARY TO HOLD MORE THAN
C         ONE SPLINE RESULT AT A TIME). IN ADDDITION, A
C         HUNT ALGORITHM 'XXHUNT' FOR SPEEDING EVALUATION
C         HAS BEEN ADDED.
C
C
C XXSPLE COMMENTS
C
C AUTHOR:   PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C           K1/0/81
C           JET EXT. 4569
C
C DATE:     14/01/91 - ADAS91: AS FOR 'XXSPLN' BUT WITH 'LINTRP()' ADDED
C
```

```
C VERSION:  1.2
C
C MODIFIED: LORNE HORTON (JET)  DATE: 25/10/97
C           - ADDED IOPT CHOICES 5, 6 AND 7
C
C VERSION:  1.3
C
C MODIFIED: Martin O'Mullane (JET)  DATE: 2/6/99
C           - SAVE nin0 and inter variables also. All compilers, ie
C             especially g77, do not automatically save (or initialise
C             variables to zero).
C
C-----------------------------------------------------------------------
C
C VERSION  : 1.1
C DATE     : 18-03-2003
C MODIFIED : Lorne Horton
C              - First version.
C
C VERSION  : 1.2
C DATE     : 10-04-2007
C MODIFIED : Allan Whiteford
C               - Modified documentation as part of automated
C   subroutine documentation preparation.
C-----------------------------------------------------------------------


C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IOPT,         NIN,          NOUT
      LOGICAL             LINTRP(NOUT),               LSETX,        LSETY
      REAL*8              D1(NIN),      D2(NIN),      D3(NIN),      DY(NIN)
      REAL*8              Q(NIN),       X(NIN),       XIN(NIN)
      REAL*8              XOUT(NOUT),   YIN(NIN),     YOUT(NOUT)
```

9.186 xxspln: Subroutine xxspln from library adaslib

```
      SUBROUTINE XXSPLN( LSETX , IOPT  , FINTX ,
     &                   NIN   , XIN   , YIN   ,
     &                   NOUT  , XOUT  , YOUT  ,
     &                   DY
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXSPLN *********************
C
C  PURPOSE:           TO INTERPOLATE/EXTRAPOLATE USING CUBIC SPLINES
C
C                     (IF IOPT < 0 NO EXTRAPOLATION TAKES PLACE = VALUES
C                      SET TO ZERO).
C
C  CALLING PROGRAMS:  GENERAL USE
C
C  SUBROUTINE:
C
C  I/O   : (L*4)  LSETX   = .TRUE.  => SET UP SPLINE PARAMETERS RELATING
C                                      TO 'XIN' AXIS.
C                           .FALSE. => DO NOT SET UP SPLINE PARAMETERS
C                                      RELATING TO 'XIN' AXIS.
C                                      (I.E. THEY WERE SET IN A PREVIOUS
C                                            CALL )
C                           ( 'LSETX' IS ALWAYS RETURN AS '.FALSE.'  ON
C                             RETURN FROM THE SUBROUTINE ).
C                           ** IMPORTANT: SEE NOTES BELOW ON 'LSETX' **
C  INPUT : (I*4)  IOPT    = SPLINE END CONDITIONS/EXTRAPOLATION CONTROL
C                           SWITCH - SEE NOTES BELOW
C                           I.E. DEFINES THE BOUNDARY DERIVATIVES.
C                                (VALID VALUES = 0, 1, 2, 3, 4)
C                           IF IOPT < 0 THEN NO EXTRAPOLATION TAKES
C                           - ANY VALUES REQUIRING EXTRAPOLATION WILL BE
C                             SET TO ZERO (END CONDITIONS AS FOR IOPT=0)
C  INPUT : (R*8)  FINTX   = INTERPOLATING X-COORDINATE TRANSFORMATION.
C                           EXTERNAL FUNCTION (SEE ROUTINES BELOW)
C
C  INPUT : (I*4)  NIN     = NUMBER OF KNOTS
C  INPUT : (R*8)  XIN()   = X-VALUES OF KNOTS
C  INPUT : (R*8)  YIN()   = Y-VALUES OF KNOTS
C
C  INPUT : (I*4)  NOUT    = NUMBER OF OUTPUT VALUES TO BE INTERPOLATED
C                           EXTRAPOLATED.
C  INPUT : (R*8)  XOUT()  = X-VALUES AT WHICH INTERPOLATION/EXTRAPOLA-
C                           TION REQUIRED
C  OUTPUT: (R*8)  YOUT()  = INTERPOLATED/EXTRAPOLATED Y-VALUES FOR
C                           REQUESTED 'XOUT()' VALUES.
C
C  OUTPUT: (R*8)  DY()    = INTERPOLATED DERIVATIVES
C
C          (I*4)  NKNOTS  = PARAMETER = MAXIMUM  NUMBER OF KNOTS ALLOWED
C          (I*4)  NIOPT   = PARAMETER = MAXIMUM  VALUE OF IOPT ALLOWED
C
C          (I*4)  I       = GENERAL ARRAY USE
C          (I*4)  K       = INDEX OF 'XOUT()' VALUE FOR INTERPOLATION/
C                           EXTRAPOLATION.
C          (I*4)  NIN0    = 'NIN' - 1
C          (I*4)  INTER   = INDEX OF CLOSEST/NEXT HIGHEST VALUE OF
C                           'XIN()' TO THE VALUE OF 'XOUT()' BEING
```

```
C                           INTERPOLATED/EXTRAPOLATED.
C          (I*4)  NOPT    = VALUE OF  'IOPT' USED IN  CALCULATING  END-
C                           CONDITIONS   FOR  STORED  'X-VALUE'  SPLINE
C                           PARAMETERS.   (NOTE:  IF  'IOPT < 0',  THEN
C                           'NOPT = 0'.) - I.E. 'NOPT = MAX( 0, IOPT )'.
C
C          (R*8)  XK      = VALUE OF 'XOUT(K)' BEING INTERPOLATED/
C                           EXTRAPOLATED
C          (R*8)  XKK     = TRANSFORMED VALUE OF 'XOUT(K)' BEING
C                           INTERPOLATED/EXTRAPOLATED.
C          (R*8)  T1      = INVERSE OF SEPARATION OF KNOTS EITHER
C                           SIDE OF CURRENT KNOT.
C          (R*8)  T2      = (CURRENT KNOT POSITION TO NEXT HIGHEST KNOT
C                            POSITION) DIVIDED BY 'T1'
C          (R*8)  T3      = (CURRENT KNOT POSITION TO NEXT LOWEST  KNOT
C                            POSITION) DIVIDED BY 'T1'
C          (R*8)  T4      = INTERPOLATION FACTOR FOR CURRENT KNOT
C          (R*8)  DL1     = (REQUESTED 'XOUT()' VALUE TO NEXT HIGHEST
C                            KNOT POSITION) DIVIDED BY SEPERATION OF
C                            KNOTS EITHER SIDE OF 'XOUT(K)'.
C          (R*8)  DL2     = (REQUESTED 'XOUT()' VALUE TO NEXT LOWEST
C                            KNOT POSITION) DIVIDED BY SEPERATION OF
C                            KNOTS EITHER SIDE OF 'XOUT(K)'.
C          (R*8)  DL2     = (REQUESTED 'XOUT()' VALUE TO NEXT LOWEST
C          (R*8)  DL3     =  SEPERATION OF KNOTS EITHER SIDE OF
C                            'XOUT(K)' * 'DL1' * 'DL2'.
C
C          (L*4)  LEXTRP  = .TRUE.  => 'EXTRAPOLATION SWITCHED ON'.
C                           .FALSE. => 'EXTRAPOLATION SWITCHED OFF'.
C
C          (R*8)  QVAL()  = VALUE OF 'Q(1)'   : FUNCTION OF 'NOPT'
C          (R*8)  D2VAL() = VALUE OF 'D2(1)'  : FUNCTION OF 'NOPT'
C          (R*8)  D3VAL() = VALUE OF 'D3(1)'  : FUNCTION OF 'NOPT'
C          (R*8)  UVAL() =  VALUE OF 'U(NIN)' : FUNCTION OF 'NOPT'
C          (R*8)  AGRL() =  POLYNOMIAL CONSTANTS FOR CUBIC SPLINE FOR
C                           GIVEN 'XOUT(K)' VALUE.
C          (R*8)  X()     =  TRANSFORMED VALUES OF 'XIN()'
C          (R*8)  H()     =  SEPERATION, ALONG X-AXIS, OF KNOT FROM NEXT
C                           HIGHEST KNOT.
C          (R*8)  Q()     =  SECOND DERIVATIVE FOR KNOT
C          (R*8)  U()     =  TEMPORARY STORAGE OF DECOMPOSED FACTORS
C          (R*8)  DELY() =  SEPERATION, ALONG Y-AXIS, OF KNOT FROM NEXT
C                           HIGHEST KNOT.
C          (R*8)  D1()    =  MULTIPLICATION FACTOR USED IN CALCULATING
C                           'U()'.
C          (R*8)  D2()    =  MULTIPLICATION FACTOR USED IN CALCULATING
C                           'U()'.
C          (R*8)  D3()    =  MULTIPLICATION FACTOR USED IN CALCULATING
C                           'U()'.
C
C          (L*4)  LUVAL()=  .TRUE. => VALUE OF 'UVAL()' REFERS TO RATE
C                                     OF CHANGE OF SLOPE AT FINAL POINT.
C                           .FALSE.=> VALUE OF 'UVAL()' REFERS TO FINAL
C                                     SLOPE
C                            FUNCTION OF 'NOPT'
C
C NOTES: 'LSETX': SET TO .TRUE. ON ENTRY IF A NEW 'XIN' ARRAY IS BEING
C                 USED.  IF THE 'XIN' AXIS IS THE SAME FOR A NUMBER OF
C                 CALLS THEN DO NOT RESET 'LSETX'  -  THIS  SUBROUTINE
C                 SETS IT TO .FALSE. FOR YOU.   IF THE VALUE OF 'NOPT'
C                 IS CHANGED BETWEEN CALLS THEN THE VALUE  OF  'LSETX'
```

```
C                 ON  ENTRY IS TAKEN AS BEING EQUAL TO .TRUE. .
C
C                 THEREFORE 'LSETX' NEED ONLY BE SET TO .TRUE. ON ENTRY
C                 IF EITHER IT IS ITS FIRST CALL OR IF ANY ONE  OF  THE
C                 FOLLOWING VALUES HAS CHANGED:
C
C                 'NIN' , 'FINTX' , 'XIN(I), I=1,NIN'
C
C                 CARE: A VARIABLE MUST BE USED FOR 'LSETX', A CONSTANT,
C                       I.E.  .TRUE. ,  CANNOT BE DIRECTLY TYPED AS  AN
C                       ARGUMENT BECAUSE IT WILL BE CHANGED TO  .FALSE.
C                       ON RETURN.
C
C          SPLINE  END CONDITIONS AND EXTRAPOLATION DEPEND ON 'IOPT' AS
C          FOLLOWS:
C
C          ------------------------------------------------------------
C          | IOPT   | NOPT |  DY(1)  DDY(1)   |  DY(N)   DDY(N)   |EXTRAP'N|
C          |--------|------|------------------|-------------------|--------|
C          | < 0    |   0  |    -     0.0     |    -      0.0     |  NO    |
C          |   0    |   0  |    -     0.0     |    -      0.0     |  YES   |
C          |   1    |   1  |    -     0.0     |  -1.5     -       |  YES   |
C          |   2    |   2  |   0.0    -       |   1.0     -       |  YES   |
C          |   3    |   3  |  -0.5    -       |  -1.5     -       |  YES   |
C          |   4    |   4  |   0.0    -       |    -      0.0     |  YES   |
C          ------------------------------------------------------------
C
C             NB. OPTIONS TO BE EXTENDED FOR POWER AND CX APPLICATION
C
C          ------------------------------------------------------------
C           IF ( IOPT.LT.0 ) - NO EXTRAPOLATION TAKES PLACE VALUES SET
C                              TO ZERO (CARE IF LOG OF OUTPUT IS NEEDED).
C           IF ( IOPT.GT.4 ) PROGRAM STOPS
C          ------------------------------------------------------------
C
C           THIS SUBROUTINE IS AN AMENDED  AND STRUCTURED VERSION OF  THE
C           SUBROUTINE  'ESPLINE'  WRITTEN BY  H.P. SUMMERS,   JET   26TH
C           OCTOBER 1989.   IT REMOVES THE COMMON BLOCK  /IONSPL/ ,   THE
C           SWITCHES 'ISW & ISW2' AND ALSO THE CASE FOR THE INTERPOLATION
C           OF CHARGE STATE VALUES.   IT INTRODUCES THE FEATURE  THAT  AN
C           ARRAY OF INPUT  'X-VALUES'  CAN BE  INTERPOLATED/EXTRAPOLATED
C           IN ONE CALL.
C
C ROUTINES:
C           ROUTINE    SOURCE    BRIEF DESCRIPTION
C           ------------------------------------------------------------
C           FINTX      ------    EXTERNAL  REAL*8  FUNCTION,   USED  TO
C                                TRANSFORM X-COORDINATES.
C
C
C AUTHOR:    PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C            K1/0/81
C            JET EXT. 4569
C
C DATE:      10/08/90 (30/08/90: IOPT = 4 ADDED & 'LUVAL' PARAMETER)
C
C UPDATE:    17/01/91 - PE BRIDEN: ADAS91 - IOPT < 0 ADDED - NO EXTRAP'N.
C                                          - NOPT DEFINITION CHANGED.
C                                          - INTRODUCED 'LEXTRP'.
C UNIX-IDL PORT:
C
```

```
C VERSION: 1.1                          DATE: 08-11-95
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST RELEASE
C
C VERSION: 1.2                          DATE: 2/6/99
C MODIFIED: Martin O'Mullane (JET)
C           - SAVE nin0 and inter variables also. All compilers, ie
C             especially g77, do not automatically save (or initialise
C             variables to zero).
C
C VERSION  : 1.3
C DATE     : 10-04-2007
C MODIFIED : Allan Whiteford
C               - Modified documentation as part of automated
C   subroutine documentation preparation.
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IOPT,        NIN,         NOUT
      LOGICAL             LSETX
      REAL*8              DY(NIN),     XIN(NIN),    XOUT(NOUT)
      REAL*8              YIN(NIN),    YOUT(NOUT)
```

9.187 xxspzd: Subroutine xxspzd from library adaslib

```
      SUBROUTINE XXSPZD( USRGRP , USRTYP , USREXT )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXSPZD *********************
C
C  PURPOSE: ADAS ROUTINE - SETS UP THE DEFAULT USEGRP, USRTYP AND USREXT
C           WHICH IDENTIFY THE FILENAME AND EXTENSION TO BE READ IN
C           SUBROUTINE SPZD.  IT WORKS IN THE SAME MANNER AS XXUID WHICH
C           WHICH ALLOWS THE DEFAULT USER SPACE TO BE SET
C
C
C           USRGRP: VALUE ON INPUT  =>  USRGRP: VALUE ON OUTPUT
C
C                     ?                 CURRENT ADAS DATA GROUPNAME
C                     *                 DEFAULT ADAS DATA GROUPNAME
C                  <BLANK>              *** USRGRP VALUE NOT CHANGED ***
C                  <OTHER>              *** USRGRP VALUE NOT CHANGED ***
C
C
C           USRTYP: VALUE ON INPUT  =>  USRTYP: VALUE ON OUTPUT
C
C                     ?                 CURRENT ADAS DATA TYPENAME
C                     *                 DEFAULT ADAS DATA TYPENAME
C                  <BLANK>              *** USRTYP VALUE NOT CHANGED ***
C                  <OTHER>              *** USRTYP VALUE NOT CHANGED ***
C
C
C           USREXT: VALUE ON INPUT  =>  USREXT: VALUE ON OUTPUT
C
C                     ?                 CURRENT ADAS DATA MEMBER EXTENS.
C                     *                 DEFAULT ADAS DATA MEMBER EXTENS
C                  <BLANK>              *** USREXT VALUE NOT CHANGED ***
C                  <OTHER>              *** USREXT VALUE NOT CHANGED ***
C
C         ? => QUERIES CURRENT ADAS DATA USRGRP, USRTYP OR USREXT
C              SETTING.
C         * => SETS ADAS DATA USEGRP, USRTYP OR USREXT SETTING
C              TO DEFAULT VALUE.
C   <BLANK> => SETS ADAS DATA USRGRP, USRTYP OR USREXT SETTING
C              TO DEFAULT VALUE.
C   <OTHER> => SETS ADAS DATA USRGRP, USRTYP OR USREXT SETTING
C              TO INPUT   VALUE.
C
C  CALLING PROGRAM: SPZD AND MAIN PROGRAMS USING SPZD
C
C  SUBROUTINE:
C
C  I/O   : (C*8)  USRGRP   = USRGRP UNDER WHICH ADAS DATA IS STORED
C                            (IF BLANK DEFAULTS TO DEFGRP)
C
CA I/O   : (C*80) USRTYP   = UNDER UNIX THIS IS OPTIONAL SUB-DIRECTORY
C                            (IF BLANK DEFAULTS TO DEFTYP)
C
C  I/O   : (C*3)  USREXT   = USREXT UNDER WHICH ADAS DATA IS STORED
C                            (IF BLANK DEFAULTS TO DEFEXT)
C
C          (C*8)  DEFGRP   = PARAMETER = DEFAULT USER GROUP FOR ADAS
C                                        DATA SOURCE
C
```

```
CA    (C*80) DEFTYP   = PARAMETER = DEFAULT SUB-DIRECTORY NAME
C                                   WHICH IS BLANK.
C
C          (C*3)  DEFEXT   = PARAMETER = DEFAULT USER EXTENSION FOR ADAS
C                                   DATA SOURCE
C
C          (C*8)  ADASGR   = CURRENT ADAS DATA SOURCE GROUP
C
CA    (C*80) ADASTY   = CURRENT SUBDIRECTORY NAME
C
C          (C*3)  ADASEX   = CURRENT ADAS DATA SOURCE EXTENSION
C
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C
C NOTE:
C          TO CHECK CURRENT ADAS SOURCE USRGRP, USRTYP AND USREXT
C          CALL XXSPZD WITH ?'S AS INPUTS.
C
C AUTHOR:  HUGH P. SUMMERS, JET
C          K1/1/67
C          JET EXT. 4941
C
C DATE:     2/09/93
C
C UPDATE:  21/11/94 - L. JALOTA - ALTERED CHARACTER STRING LENGTHS FOR UNIX
C
C-----------------------------------------------------------------------
      CHARACTER*3         USREXT
      CHARACTER*8         USRGRP
      CHARACTER*80        USRTYP
```

9.188 xxssxb: Subroutine xxssxb from library adaslib

```
      SUBROUTINE XXSSXB( USRGRP , USRTYP , USREXT )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXSSXB *********************
C
C  PURPOSE: ADAS ROUTINE - SETS UP THE DEFAULT USRGRP, USRTYP AND USREXT
C           WHICH IDENTIFY THE FILENAME AND EXTENSION TO BE READ IN
C           SUBROUTINE SSXB.  IT WORKS IN THE SAME MANNER AS XXUID WHICH
C           WHICH ALLOWS THE DEFAULT USER SPACE TO BE SET
C
C
C           USRGRP: VALUE ON INPUT  =>  USRGRP: VALUE ON OUTPUT
C
C                     ?                 CURRENT ADAS DATA GROUPNAME
C                     *                 DEFAULT ADAS DATA GROUPNAME
C                  <BLANK>              *** USRGRP VALUE NOT CHANGED ***
C                  <OTHER>              *** USRGRP VALUE NOT CHANGED ***
C
C
C           USRTYP: VALUE ON INPUT  =>  USRTYP: VALUE ON OUTPUT
C
C                     ?                 CURRENT ADAS DATA TYPENAME
C                     *                 DEFAULT ADAS DATA TYPENAME
C                  <BLANK>              *** USRTYP VALUE NOT CHANGED ***
C                  <OTHER>              *** USRTYP VALUE NOT CHANGED ***
C
C
C           USREXT: VALUE ON INPUT  =>  USREXT: VALUE ON OUTPUT
C
C                     ?                 CURRENT ADAS DATA MEMBER EXTENS.
C                     *                 DEFAULT ADAS DATA MEMBER EXTENS
C                  <BLANK>              *** USREXT VALUE NOT CHANGED ***
C                  <OTHER>              *** USREXT VALUE NOT CHANGED ***
C
C         ? => QUERIES CURRENT ADAS DATA USRGRP, USRTYP OR USREXT
C              SETTING.
C         * => SETS ADAS DATA USRGRP USRTYP OR USREXT SETTING
C              TO DEFAULT VALUE.
C   <BLANK> => SETS ADAS DATA USRGRP, USRTYP OR USREXT SETTING
C              TO DEFAULT VALUE.
C   <OTHER> => SETS ADAS DATA USRGRP, USRTYP OR USREXT SETTING
C              TO INPUT   VALUE.
C
C  CALLING PROGRAM: SSXB AND MAIN PROGRAMS USING SSXB
C
C  SUBROUTINE:
C
CA I/O   : (C*8)  USRGRP   = USRGRP UNDER WHICH ADAS DATA IS STORED
C                            (IF BLANK DEFAULTS TO DEFGRP)
C
CA I/O   : (C*80  USRTYP   = OPTIONAL SUB-DIRECTORY UNDER UNIX
C                            (IF BLANK DEFAULTS TO DEFTYP)
C
C  I/O   : (C*3)  USREXT   = USREXT UNDER WHICH ADAS DATA IS STORED
C                            (IF BLANK DEFAULTS TO DEFEXT)
C
CA         (C*8)  DEFGRP   = PARAMETER = DEFAULT USER GROUP FOR ADAS
C                                        DATA SOURCE
C
```

```
CA         (C*80) DEFTYP   = PARAMETER = DEFAULT USER TYPE FOR ADAS
C                                        DATA SOURCE
C
C          (C*3)  DEFEXT   = PARAMETER = DEFAULT USER EXTENSION FOR ADAS
C                                        DATA SOURCE
C
CA         (C*8)   ADASGR   = CURRENT ADAS DATA SOURCE GROUP
C
CA         (C*80)  ADASTY   = CURRENT SUB-DIRECTORY NAME ( OPTIONAL)
C
C          (C*3)  ADASEX   = CURRENT ADAS DATA SOURCE EXTENSION
C
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C
C NOTE:
C          TO CHECK CURRENT ADAS SOURCE USRGRP, USRTYP AND USREXT
C          CALL XXSSXB WITH '?' AS INPUTS.
C
C AUTHOR:  HUGH P. SUMMERS, JET
C          K1/1/57
C          JET EXT. 4941
C
C DATE:     2/09/93
C
C UPDATES: 24/10/94  L. JALOTA (TESSELLA SUPPORT SERVICES PLC)
C      CHANGED VALUES OF DEFGRP,DEFTYP, DEFEXT SUITABLE
C      FOR DEC ALPHA DIRECTORY STRUCTURE.
C UPDATE : 22/11/94  L. JALOTA  -  TIDIED UP CHARACTER LENGTH DEFINITIONS.
C
C-----------------------------------------------------------------------
      CHARACTER*3         USREXT
      CHARACTER*8         USRGRP
      CHARACTER*80        USRTYP
```

9.189 xxsszd: Subroutine xxsszd from library adaslib

```
      SUBROUTINE XXSSZD( USRGRP , USRTYP , USREXT )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXSSZD *********************
C
C  PURPOSE: ADAS ROUTINE - SETS UP THE DEFAULT USEGRP, USRTYP AND USREXT
C           WHICH IDENTIFY THE FILENAME AND EXTENSION TO BE READ IN
C           SUBROUTINE SSZD.  IT WORKS IN THE SAME MANNER AS XXUID WHICH
C           WHICH ALLOWS THE DEFAULT USER SPACE TO BE SET
C
C
C           USRGRP: VALUE ON INPUT  =>  USRGRP: VALUE ON OUTPUT
C
C                     ?                 CURRENT ADAS DATA GROUPNAME
C                     *                 DEFAULT ADAS DATA GROUPNAME
C                  <BLANK>              *** USRGRP VALUE NOT CHANGED ***
C                  <OTHER>              *** USRGRP VALUE NOT CHANGED ***
C
C
C           USRTYP: VALUE ON INPUT  =>  USRTYP: VALUE ON OUTPUT
C
C                     ?                 CURRENT ADAS DATA TYPEPNAME
C                     *                 DEFAULT ADAS DATA TYPEGROUPNAME
C                  <BLANK>              *** USRTYP VALUE NOT CHANGED ***
C                  <OTHER>              *** USRTYP VALUE NOT CHANGED ***
C
C
C           USREXT: VALUE ON INPUT  =>  USREXT: VALUE ON OUTPUT
C
C                     ?                 CURRENT ADAS DATA MEMBER EXTENS.
C                     *                 DEFAULT ADAS DATA MEMBER EXTENS
C                  <BLANK>              *** USREXT VALUE NOT CHANGED ***
C                  <OTHER>              *** USREXT VALUE NOT CHANGED ***
C
C         ? => QUERIES CURRENT ADAS DATA USRGRP, USRTYP OR USREXT
C              SETTING.
C         * => SETS ADAS DATA USEGRP, USRTYP OR USREXT SETTING
C              TO DEFAULT VALUE.
C   <BLANK> => SETS ADAS DATA USRGRP, USRTYP OR USREXT SETTING
C              TO DEFAULT VALUE.
C   <OTHER> => SETS ADAS DATA USRGRP, USRTYP OR USREXT SETTING
C              TO INPUT   VALUE.
C
C  CALLING PROGRAM: SSZD AND MAIN PROGRAMS USING SSZD
C
C  SUBROUTINE:
C
C I/O   : (C*8)  USRGRP   = USRGRP UNDER WHICH ADAS DATA IS STORED
C                           (IF BLANK DEFAULTS TO DEFGRP)
C
CA I/O   : (C*80) USRTYP   = OPTIONAL SUBDIRECTORY FOR FILE
C                           (IF BLANK DEFAULTS TO DEFTYP)
C
C                           (IF BLANK DEFAULTS TO DEFEXT)
C
C          (C*8)  DEFGRP   = PARAMETER = DEFAULT USER GROUP FOR ADAS
C                                        DATA SOURCE
C
CA         (C*80) DEFTYP   = OPTIONAL SUB-DIRECTORY DEFAULT IS BLANK
```

```
C                                       DATA SOURCE
C
C          (C*3)  DEFEXT   = PARAMETER = DEFAULT USER EXTENSION FOR ADAS
C                                       DATA SOURCE
C
C          (C*8)  ADASGR   = CURRENT ADAS DATA SOURCE GROUP
C
CA         (C*80) ADASTY   = OPTIONAL SUB-DIRECTORY
C
C          (C*3)  ADASEX   = CURRENT ADAS DATA SOURCE EXTENSION
C
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C
C NOTE:
C          TO CHECK CURRENT ADAS SOURCE USRGRP, USRTYP AND USREXT
C          CALL XXSSZD WITH ?'S AS INPUTS.
C
C AUTHOR:  HUGH P. SUMMERS, JET
C          K1/1/67
C          JET EXT. 4941
C
C DATE:     2/09/93
C
C UPDATE:  L. JALOTA - 10/11/94 : MODIFIED FOR USE UNDER UNIX
C   INCREASED STRING LENGTHS
C UPDATE:  L. JALOTA - 22/11/94 : TIDIED UP CHARACTER LENGTH DEFINITIONS.
C-----------------------------------------------------------------------
      CHARACTER*3         USREXT
      CHARACTER*8         USRGRP
      CHARACTER*80        USRTYP
```

9.190 xxstnp: Subroutine xxstnp from library adaslib

```
      SUBROUTINE XXSTNP( IUNIT , PGLEN  , NBLOCK , NLINES )
C-----------------------------------------------------------------------
C
C  ******************** FORTRAN77 SUBROUTINE: XXSTNP ******************
C
C  PURPOSE: CHECK WHETHER OUTPUT BLOCK WILL FIT ON CURRENT PAGE. IF NOT
C           A NEW PAGE IS INITIATED, AND LINE COUNT RESET ACCORDINGLY.
C
C  CALLING PROGRAM: GENERAL USE
C
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  IUNIT   = OUTPUT UNIT FOR RESULTS
C  INPUT : (I*4)  PGLEN   = OUTPUT PAGE LENGTH IN LINES
C  INPUT : (I*4)  NBLOCK  = NUMBER OF LINES IN BLOCK
C  I/O   : (I*4)  NLINES  = LAST PAGE LINE WRITTEN TOO.
C
C ROUTINES: NONE
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 4569
C
C DATE:    05/07/90
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             IUNIT,       NBLOCK,      NLINES,      PGLEN
```

9.191 xxstuc: Subroutine xxstuc from library adaslib

```
      SUBROUTINE XXSTUC( STRING )
C
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXSTUC *********************
C
C  PURPOSE: ENSURES ALL LETTERS IN INPUT STRING ARE UPPER CASE.
C
C  CALLING PROGRAM: GENERAL USE.
C
C  I/O   : (C*(*)) STRING = STRING.
C
C          (I*4)  IDIFF   = DIFFERENCE IN CODES BETWEEN LOWER AND UPPER
C                           CASE LETTER.
C          (I*4)  I       = LOOP INDEX.
C
C ROUTINES: NONE
C
C AUTHOR:  JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 5183
C
C DATE:    26/11/93
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      CHARACTER*(*)       STRING
```

9.192 xxtcon: Subroutine xxtcon from library adaslib

```
      SUBROUTINE XXTCON( INTYP, OUTTYP, IZ1, ITVAL, TIN, TOUT )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXTCON *********************
C
C  PURPOSE: TO CONVERT AN ARRAY OF TEMPERATURES INTO SPECIFIED UNITS
C
C  CALLING PROGRAM: GENERAL USE
C
C  SUBROUTINE:
C
C  INPUT :    (I*4)  INTYP   = 1 => 'TIN(array)' UNITS: KELVIN
C                              = 2 => 'TIN(array)' UNITS: eV
C                              = 3 => 'TIN(array)' UNITS: REDUCED TEMP.
C  INPUT :    (I*4)  OUTTYP  = 1 => 'TOUT(array)' UNITS: KELVIN
C                              = 2 => 'TOUT(array)' UNITS: eV
C                              = 3 => 'TOUT(array)' UNITS: REDUCED TEMP.
C  INPUT :    (I*4)  IZ1     = RECOMBINING ION CHARGE (= Z+1).
C  INPUT :    (I*4)  ITVAL   = NUMBER OF TEMPERATURES IN 'TIN(array)'
C  INPUT :    (R*8)  TIN()   = INPUT TEMPERATURES (STATED UNITS)
C  OUTPUT:    (R*8)  TOUT()  = OUTPUT TEMPERATURES (STATED UNITS)
C
C             (R*8)  EV2KEL  = ELECTRON VOLTS TO KELVIN CONVERSION
C             (R*8)  KEL2EV  = KELVIN TO ELECTRON VOLTS CONVERSION
C
C             (I*4)  I       = GENERAL USE
C
C             (R*8)  Z1P2    = 'IZ1'**2
C             (R*8)  TCONV() = TEMPERATURE CONVERSION PARAMETERS
C
C ROUTINES:  NONE
C
C NOTE:
C            TEMPERATURE CONVERSION PARAMETERS:
C
C            INTYP = 1 ; TCONV(1) =             KELVIN  -> OUTPUT UNITS
C            INTYP = 2 ; TCONV(2) =                 eV  -> OUTPUT UNITS
C            INTYP = 3 ; TCONV(3) => REDUCED TEMPERATURE -> OUTPUT UNITS
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 4569
C
C DATE:    04/01/91
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER             INTYP,       ITVAL,       IZ1,         OUTTYP
      REAL*8              TIN(ITVAL),  TOUT(ITVAL)
```

9.193 xxterm: Subroutine xxterm from library adaslib

```
      SUBROUTINE XXTERM
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXTERM *********************
C
C  PURPOSE: TERMINATES PROGRAM WITH MESSAGE.
C
C  CALLING PROGRAM: GENERAL USE
C
C  SUBROUTINE: NO VARIABLES
C
C ROUTINES: NONE
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 4569
C
C DATE:    17/08/90
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      WRITE(0,1000)
      STOP
C-----------------------------------------------------------------------
 1000 FORMAT(/1X,29('*'),' PROGRAM TERMINATED ',29('*'))
C-----------------------------------------------------------------------
      END
```

9.194 xxtoday: Subroutine xxtoday from library adaslib

```
      subroutine xxtoday(str_today)

C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: xxtoday ********************
C
C PURPOSE: To return today's date.
C
C CALLING PROGRAM: General use
C
C OUTPUT: (C*10)  str_today    = Today's data in DD-MM-YYYY format.
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          TODAY      ADAS      Wrapper to C library routine localtime
C
C
C AUTHOR   : Martin O'Mullane
C DATE     : 19-04-2005
C
C
C VERSION  : 1.1
C DATE     : 19-04-2005
C MODIFIED : Martin O'Mullane
C               - First version.
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      external      today
C-----------------------------------------------------------------------
      CHARACTER*10        STR_TODAY
```

9.195 xxuid: Subroutine xxuid from library adaslib

```
      SUBROUTINE XXUID( USERID )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXUID  *********************
C
C  PURPOSE: ADAS ROUTINE - SETS UP THE DEFAULT USERID WHICH STORES THE
C           DATA TO BE READ USING STANDARD ADAS DATA READING ROUTINES.
CA     UNDER UNIX PORT THIS NOW GETS THE ENVIRONMENT VARIABLE
CA     "ADASCENT".
C
C           USERID: VALUE ON INPUT  =>  USERID: VALUE ON OUTPUT
C
C                     ?                 CURRENT ADAS DATA SOURCE USER ID
C                     *                 DEFAULT ADAS DATA SOURCE USER ID
C                  <BLANK>              *** USERID VALUE NOT CHANGED ***
C                  <OTHER>              *** USERID VALUE NOT CHANGED ***
C
C           ? => QUERIES CURRENT ADAS SOURCE USERID SETTING.
C           * => SETS ADAS SOURCE USERID SETTING TO DEFAULT VALUE.
C     <BLANK> => SETS ADAS SOURCE USERID SETTING TO DEFAULT VALUE.
C     <OTHER> => SETS ADAS SOURCE USERID SETTING TO INPUT   VALUE.
C
C  CALLING PROGRAM: GENERAL USE
C
C  SUBROUTINE:
C
C
C  I/O   : (C*80)  USERID   = USERID UNDER WHICH ADAS DATA IS STORED
C                            (IF BLANK DEFAULTS TO DEFUID)
C
C          (C*80)  DEFUID   = PARAMETER = DEFAULT USER ID FOR ADAS DATA
C                                         SOURCE
C
C          (C*80)  ADASID   = CURRENT ADAS DATA SOURCE USER ID
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C
C NOTE:
C          TO CHECK CURRENT ADAS SOURCE USERID CALL XXUID WITH
C          ? AS INPUT.
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/37
C          JET EXT. 5023
C
C DATE:    10/03/93
C
C UPDATE:  L. JALOTA  - 7/11/94 : ADDED CALL TO GETENV TO FETCH UNIX
C   ENVIRONMENT VARIABLE ADASUSER
C
C VERSION: 1.2                          DATE: 08-08-96
C MODIFIED: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C               - REMOVED SUPERFLUOUS 'EXTERNAL GETENV' STATEMENT
C
C-----------------------------------------------------------------------
      CHARACTER*80        USERID
```

9.196 xxwcmt_15: Subroutine xxwcmt_15 from library adaslib

```
       subroutine xxwcmt_15( iunit      ,
     &                       ndstore    , ndcmt    , ndfld   , ndsyn   ,
     &                       ndion      , ndopt    , ndlev   ,
     &                       lroot      , lsuper   ,
     &                       nfld       , isyn     , fldk    ,
     &                       lfld       ,
     &                       iz0        , iz1      , popcode ,
     &                       dsn04      , dsn18    ,
     &                       nion       , cion     , lion    ,
     &                       nopt       , copt     , lopt    ,
     &                       nlev       , config   , cterm   , ener    ,
     &                       esym       , dsnpt    ,
     &                       iptnl      , is       , tabul   , units   ,
     &                       nbsel      , nspb     , nspp    ,
     c                       ntrans     , ctrans   , wtrans  ,
     &                       ctype      , ispbr    , isppr   , iszr    ,
     &                       itg        , ipr      , iwr     ,
     &                       code       , producer , date
     &                     )
c-----------------------------------------------------------------------
c
c ****************** fortran 77 subroutine: xxwcmt_15 ******************
c
c  purpose: To write the comment section of an adf15 file .
c
c  notes:  (1) The  routine checks for the presence of field keys of
c              their synonyms, returns a value for the field if
c              present and the location & range of comment lines
c              associated with the field key if appropriate.  This
c              follows the general pattern of adf comment reading
c              subroutines.
c          (2) Analysis is carried out on the associated comment lines
c              of specific field keys to isolate and return relevant
c              information.  This is specific to the adf number.  These
c              returned data should match the data which needs to be
c              provided to enable writing of identical comment lines
c              by 'xxwcmt_15.for'
c
c  subroutine:
c
c  input : (i*4)  iunit      = unit number for input adf15 file
c  input : (i*4)  ndcnct     = maximum number of elements in connection
c  input : (i*4)  ndcmt      = maximum number of comment text lines
c  input : (i*4)  ndfld      = maximum number of search field
c  input : (i*4)  ndion      = maximum number of selective ionis coefft.
c                              inclusions
c  input : (i*4)  ndopt      = maximum number of options keys set in
c                              population code
c  input : (i*4)  ndlev      = maximum number of levels included in
c                              population calculation

c  input : (l*4)  lroot      = .true.  => output root comments
c                              .false. => do not output root comments
c  input : (l*4)  lsuper     = .true.  => output superstage comments
c                              .false. => do not output superstage comments
c  input : (i*4)  nfld       = number of fields for adf15 comments
c  input : (i*4)  isyn()     = number of synonyms for fields
c                              1st dim: field index (1->nfld)
c  input :  (c*40) fldk(,)    = field keys
```

```
c                                1st dim: field index (1->nfld)
c                                2nd dim: synonymn index (1->isyn())
c  input : (l*4)  lfld()      = .true.  => field  available for comments
c                              = .false. => field not available
c                                1st dim: field index (1->nfld)
c  input : (i*4)  iz0          = nuclear charge
c  input : (i*4)  iz1          = emitting ion charge+1
c  input : (c*7)  popcode      = propulation processing code
c  input : (c*120)dsn04        = adf04 file used by population code
c  input : (c*120)dsn18        = adf18 map file used to access projection
c  input : (i*4)  nion         = number of selective ionis. coefft.
c                               inclusions in population calculation
c  input : (c*5)  cion()       = selec. ionis coefft. spec as (ispb,ispp)
c                                1st dim: ionis coefft. list index
c  input : (l*1)  lion()       = .true. => included
c                              = .false.=> not included
c                                1st dim: ionis coefft. list index
c  input : (i*4)  nopt         = number of option keys present for
c                                population calculation
c  input : (c*6)  copt()       = option specification strings as l*****
c                                1st dim: option list index
c  input : (l*1)  lopt()       = .true. => set
c                              = .false.=> not set
c                                1st dim: option list index
c  input : (i*4)  nlev         = number of energy levels included in
c                                population calculation
c  input : (c*19) config()     = configuration string
c                                1st dim: level list index
c  input : (c*14) cterm()      = term/level specification string
c                                1st dim: level list index
c  input : (r*8)  ener()       = energy level relative to lowest (cm^-1)
c                                1st dim: level list index
c  input : (c*2)  esym         = element symbol
c  input : (c*120)dsnpt        = parent file template used to create
c                                current child partition (blank if root)
c  input : (i*4)  iptnl        = current partition level
c  input : (i*4)  is           = superstage label
c  input : (c*40) tabul        = adf15 quantity tabulated specification
c  input : (c*40) units        = adf15 units use specification
c  input : (i*4)  nspb         = number of excitation (base) drivers for
c                                superstage (= icnctv(is))
c  input : (i*4)  nspp         = number of recombination (parent) drivers
c                                for superstage (=icnctv(is+1))
c  input : (i*4)  nbsel        = number of emissivity line blocks in the
c                                adf15 file
c  input : (i*4)  ntrans       = number of distinct transitions in the
c                                adf15 file
c  input : (c*29) ctrans()     = transition specification string
c                                1st dim: emissivity index (1->ntrans)
c  input : (r*8)  wtrans()     = transition wavelength
c
c  input : (c*5)  ctype()      = transition type for each line block
c                                1st dim: index of block in adf15 file
c  input : (i*4)  ispbr()      = base driver index for each line block
c                                1st dim: index of block in adf15 file
c  input : (i*4)  isppr()      = parent driver index for each line block
c                                1st dim: index of block in adf15 file
c  input : (i*4)  iszr()       = ion charge relating to each line
c                                1st dim: index of block in adf15 file
c  input : (i*4)  itg()        = transition group attribution of
c                                emissivity  line block
```

```
c                              1st dim: index of block in adf15 file
c  input : (i*4)  ipr()      = power ranking of emissivity line
c                              block (note power is a composite
c                              attribute of a transition group)
c                              1st dim: index of block in adf15 file
c  input : (c*7)  code       = ADAS code which generated the
c                              superstage adf15 file
c  input : (c*30) producer   = producer of the adf15 file
c  input : (c*8)  date       = date of creattion of the superstage
c                              adf15 file
c
c routines:
c          routine    source    brief description
c          ------------------------------------------------------------
c          i4unit     adas      fetch unit number for output of messages
c          r8fctn     adas      convert string to real number
c          xxslen     adas      find string less front and tail blanks
c          xxcase     adas      convert a string to upper or lower case
c          xxordr     adas      order a real vector retaining indexing
c
c
c author:  H. P. Summers, University of Strathclyde
c          ja7.08
c          tel. 0141-548-4196
c
c date:    19/05/06
c
c
c version  : 1.1
c date     : 19-05-2006
c modified : H P Summers
c             - first version.
c
c-----------------------------------------------------------------------
      CHARACTER*5         CION(NDION)
      CHARACTER*7         CODE
      CHARACTER*19        CONFIG(NDLEV)
      CHARACTER*6         COPT(NDOPT)
      CHARACTER*14        CTERM(NDLEV)
      CHARACTER*29        CTRANS(NDSTORE)
      CHARACTER*5         CTYPE(NDSTORE)
      CHARACTER*8         DATE
      CHARACTER*80        DSN04,       DSN18
      CHARACTER*120       DSNPT
      CHARACTER*2         ESYM
      CHARACTER*40        FLDK(NDFLD,NDSYN)
      CHARACTER*7         POPCODE
      CHARACTER*30        PRODUCER
      CHARACTER*60        TABUL,       UNITS
      INTEGER             IPR(NDSTORE),             IPTNL,       IS
      INTEGER             ISPBR(NDSTORE),           ISPPR(NDSTORE)
      INTEGER             ISYN(NDFLD), ISZR(NDSTORE)
      INTEGER             ITG(NDSTORE),             IUNIT
      INTEGER             IWR(NDSTORE),             IZ0,         IZ1
      INTEGER             NBSEL,       NDCMT,       NDFLD,       NDION
      INTEGER             NDLEV,       NDOPT,       NDSTORE,     NDSYN
      INTEGER             NFLD,        NION,        NLEV,        NOPT
      INTEGER             NSPB,        NSPP,        NTRANS
      LOGICAL             LFLD(NDFLD), LION(NDION), LOPT(NDOPT), LROOT
      LOGICAL             LSUPER
      REAL*8              ENER(NDLEV), WTRANS(NDSTORE)
```

9.197 xxwcmt_40: Subroutine xxwcmt_40 from library adaslib

```
       subroutine xxwcmt_40( iunit      ,
     &                       ndstore    , ndcmt    , ndfld   , ndsyn   ,
     &                       ndion      , ndopt    , ndlev   ,
     &                       lroot      , lsuper   ,
     &                       nfld       , isyn     , fldk    ,
     &                       lfld       ,
     &                       iz0        , iz1      , popcode ,
     &                       dsn04      , dsn18    ,
     &                       nion       , cion     , lion    ,
     &                       nopt       , copt     , lopt    ,
     &                       nlev       , config   , cterm   , ener    ,
     &                       esym       , dsnpt    ,
     &                       iptnl      , is       , tabul   , units   ,
     &                       nbsel      , nspb     , nspp    ,
     &                       nrange     , npixr    , wvminr  , wvmaxr  ,
     &                       ilzr       , ihzr     ,
     &                       ctype      , ispbr    , isppr   ,
     &                       irg        , ipr      , iwr     ,
     &                       code       , producer , date
     &                     )
c-----------------------------------------------------------------------
c
c ****************** fortran 77 subroutine: xxwcmt_40 ******************
c
c  purpose: To write the comment section of an adf40 file .
c
c  notes:  (1) The  routine checks for the presence of field keys of
c              their synonyms, returns a value for the field if
c              present and the location & range of comment lines
c              associated with the field key if appropriate.  This
c              follows the general pattern of adf comment reading
c              subroutines.
c          (2) Analysis is carried out on the associated comment lines
c              of specific field keys to isolate and return relevant
c              information.  This is specific to the adf number.  These
c              returned data should match the data which needs to be
c              provided to enable writing of identical comment lines
c              by 'xxwcmt_15.for'
c
c  subroutine:
c
c  input : (i*4)  iunit      = unit number for input adf40 file
c  input : (i*4)  ndstore    = maximum number  of  input data-blocks
c                              that can be stored
c  input : (i*4)  ndcmt      = maximum number of comment text lines
c  input : (i*4)  ndfld      = maximum number of search field
c  input : (i*4)  ndion      = maximum number of selective ionis coefft.
c                              inclusions
c  input : (i*4)  ndopt      = maximum number of options keys set in
c                              population code
c  input : (i*4)  ndlev      = maximum number of levels included in
c                              population calculation
c
c  input : (l*4)  lroot      = .true.  => output root comments
c                              .false. => do not output root comments
c  input : (l*4)  lsuper     = .true.  => output superstage comments
c                              .false. => do not output superstage comments
c  input : (i*4)  nfld       = number of fields for adf40 comments
c  input : (i*4)  isyn()     = number of synonyms for fields
```

```
c                                1st dim: field index (1->nfld)
c  input :  (c*40) fldk(,)     = field keys
c                                1st dim: field index (1->nfld)
c                                2nd dim: synonymn index (1->isyn())
c  input : (l*4)  lfld()       = .true.  => field  available for comments
c                               = .false. => field not available
c                                1st dim: field index (1->nfld)
c  input : (i*4)  iz0           = nuclear charge
c  input : (i*4)  iz1           = emitting ion charge+1
c  input : (c*7)  popcode       = population processing code
c  input : (c*120)dsn04         = adf04 file used by population code
c  input : (c*120)dsn18         = adf18 map file used to access projection
c  input : (i*4)  nion          = number of selective ionis. coefft.
c                                inclusions in population calculation
c  input : (c*5)  cion()        = selec. ionis coefft. spec as (ispb,ispp)
c                                1st dim: ionis coefft. list index
c  input : (l*1)  lion()        = .true. => included
c                               = .false.=> not included
c                                1st dim: ionis coefft. list index
c  input : (i*4)  nopt          = number of option keys present for
c                                population calculation
c  input : (c*6)  copt()        = option specification strings as l*****
c                                1st dim: option list index
c  input : (l*1)  lopt()        = .true. => set
c                               = .false.=> not set
c                                1st dim: option list index
c  input : (i*4)  nlev          = number of energy levels included in
c                                population calculation
c  input : (c*19) config()      = configuration string
c                                1st dim: level list index
c  input : (c*14) cterm()       = term/level specification string
c                                1st dim: level list index
c  input : (r*8)  ener()        = energy level relative to lowest (cm^-1)
c                                1st dim: level list index
c  input : (c*2)  esym          = element symbol
c  input : (c*120)dsnpt         = parent file template used to create
c                                current child partition (blank if root)
c  input : (i*4)  iptnl         = current partition level
c  input : (i*4)  is            = superstage label
c  input : (c*40) tabul         = adf40 quantity tabulated specification
c  input : (c*40) units         = adf40 units use specification
c  input : (i*4)  nspb          = number of excitation (base) drivers for
c                                superstage (= icnctv(is))
c  input : (i*4)  nspp          = number of recombination (parent) drivers
c                                for superstage (=icnctv(is+1))
c  input : (i*4)  nbsel         = number of emissivity line blocks in the
c                                adf40 file
c  input : (i*4)  nrange        = number of distinct wave length ranges
c                                in the  adf40 file
c
c  input : (i*4)  npixr()       = number of pivels in wavelength range
c                                1st dim: wavelength range count
c  input : (r*8)  wvminr()      = min. wavelength (A) of wavelength range
c                                1st dim: wavelength range count
c  input : (r*8)  wvmaxr()      = max. wavelength (A) of wavelength range
c                                1st dim: wavelength range count
c  input : (i*4)  ilzr()        = lowest charge state contributing to
c                                the wavelength range
c                                1st dim: wavelength range count
c  input : (i*4)  ihzr()        = highest charge state contributing to
c                                the wavelength range
```

```
c                               1st dim: wavelength range count
c  input : (c*5)  ctype()    = transition type for each line block
c                               1st dim: index of block in adf40 file
c  input : (i*4)  ispbr()    = base driver index for each line block
c                               1st dim: index of block in adf40 file
c  input : (i*4)  isppr()    = parent driver index for each line block
c                               1st dim: index of block in adf40 file
c  input : (i*4)  iszr()     = ion charge relating to each line
c                               1st dim: index of block in adf40 file
c  input : (i*4)  irg()      = transition group attribution of
c                               emissivity  line block
c                               1st dim: index of block in adf40 file
c  input : (i*4)  ipr()      = power ranking of emissivity line
c                               block (note power is a composite
c                               attribute of a transition group)
c                               1st dim: index of block in adf40 file
c  input : (c*7)  code       = ADAS code which generated the
c                               superstage adf40 file
c  input : (c*30) producer   = producer of the adf40 file
c  input : (c*8)  date       = date of creattion of the superstage
c                               adf40 file
c
c routines:
c          routine    source    brief description
c          ------------------------------------------------------------
c          i4unit     adas      fetch unit number for output of messages
c          r8fctn     adas      convert string to real number
c          xxslen     adas      find string less front and tail blanks
c          xxcase     adas      convert a string to upper or lower case
c          xxordr     adas      order a real vector retaining indexing
c
c
c author:  h. p. summers, university of strathclyde
c          ja7.08
c          tel. 0141-548-4196
c
c date:    15/06/06
c
c
c version  : 1.1
c date     : 15-06-2006
c modified : H P Summers
c             - first version.
c
c-----------------------------------------------------------------------
      CHARACTER*5         CION(NDION)
      CHARACTER*7         CODE
      CHARACTER*19        CONFIG(NDLEV)
      CHARACTER*6         COPT(NDOPT)
      CHARACTER*14        CTERM(NDLEV)
      CHARACTER*5         CTYPE(NDSTORE)
      CHARACTER*8         DATE
      CHARACTER*80        DSN04,       DSN18
      CHARACTER*120       DSNPT
      CHARACTER*2         ESYM
      CHARACTER*40        FLDK(NDFLD,NDSYN)
      CHARACTER*7         POPCODE
      CHARACTER*30        PRODUCER
      CHARACTER*60        TABUL,       UNITS
      INTEGER             IHZR(NDSTORE),            ILZR(NDSTORE)
      INTEGER             IPR(NDSTORE),             IPTNL
```

```
INTEGER             IRG(NDSTORE),              IS
INTEGER             ISPBR(NDSTORE),            ISPPR(NDSTORE)
INTEGER             ISYN(NDFLD), IUNIT,        IWR(NDSTORE)
INTEGER             IZ0,         IZ1,          NBSEL,        NDCMT
INTEGER             NDFLD,       NDION,        NDLEV,        NDOPT
INTEGER             NDSTORE,     NDSYN,        NFLD,         NION
INTEGER             NLEV,        NOPT,         NPIXR(NDSTORE)
INTEGER             NRANGE,      NSPB,         NSPP
LOGICAL             LFLD(NDFLD), LION(NDION), LOPT(NDOPT), LROOT
LOGICAL             LSUPER
REAL*8              ENER(NDLEV), WVMAXR(NDSTORE)
REAL*8              WVMINR(NDSTORE)
```

9.198 xxwlgc: Subroutine xxwlgc from library adaslib

```
      SUBROUTINE XXwLGC ( LVAR )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXWLGC *********************
C
C  PURPOSE: Sends a 1 for true or a 0 for false to IDL
C
C  CALLING PROGRAM: GENERAL USE
C
C
C  SUBROUTINE:
C
C
C INPUT:   (L)     LVAT     = Logic variable to write to IDL
C
C          (I*4)   PIPEOU   = PARAMETER = UNIT NUMBER FOR INPUT TO PIPE
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          XXFLSH     ADAS      FLUSHES I/O BUFFER
C
C AUTHOR:  Martin O'Mullane
C
C DATE:    28-02-2003
C
C-----------------------------------------------------------------------
      LOGICAL             LVAR
```

9.199 xxword: Subroutine xxword from library adaslib

```
      SUBROUTINE XXWORD( CTEXT  , CDELIM , NFIRST ,
     &                   IWORDS ,
     &                   IFIRST , ILAST  , NWORDS
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXWORD *********************
C
C  PURPOSE: TO EXTRACT THE Nfirst to (Nfirst+IWORDS-1) WORDS FROM AN
C           INPUT STRING. OUTPUTS THE FIRST AND LAST BYTE INDEXES OF
C           EACH WORD AS WELL AS THE TOTAL NUMBER OF WORDS FOUND.
C
C           A WORD = A STRING OF CHARACTERS SEPARATED BY ANY CHARACTER
C                    CONTAINED IN THE INPUT STRING CDELIM.
C
C  CALLING PROGRAM: GENERAL USE
C
C  SUBROUTINE:
C
C  INPUT : (C*(*)) CTEXT   = INPUT TEXT LINE CONTAINING STRING
C  INPUT : (C*(*)) CDELIM  = INPUT STRING CONTAINING DELIMITER CHARS.
C  INPUT : (I*4)   NFIRST  = THE INDEX NO. OF THE FIRST WORD TO EXTRACT.
C
C  I/O   : (I*4)   IWORDS  = INPUT : SIZE OF IFIRST, ILAST(ARRAYS)
C                                    (I.E. NUMBER OF WORDS TO EXTRACT)
C                            = OUTPUT: NUMBER OF REQUESTED WORDS FOUND
C
C  OUTPUT: (I*4)   IFIRST()= INDEX OF FIRST BYTE OF THE Nth WORD
C  OUTPUT: (I*4)   ILAST() = INDEX OF LAST  BYTE OF THE Nth WORD
C  OUTPUT: (I*4)   NWORDS  = THE TOTAL NUMBER OF WORDS FOUND IN CTEXT
C
C          (I*4)   LENTXT  = LENGTH IN BYTES OF 'CTEXT' STRING
C          (I*4)   IDELIM  = 0 => CTEXT CHARACTER IS NOT A DELIMITER
C                          > 0 => CTEXT CHARACTER IS A DELIMITER
C          (I*4)   ITOTAL  = NUMBER OF WORDS FOUND SO FAR
C          (I*4)   IINDEX  = IFIRST()/ILAST() INDEX OF CURRENT WORD
C          (I*4)   NLAST   = THE INDEX NO. OF THE LAST WORD TO EXTRACT
C          (I*4)   I       = GENERAL USE INDEX
C
C          (L*4)   LWORD   = .TRUE.  - PROCESSING AN IDENTIFIED WORD
C                            .FALSE. - PROCESSING SPACE BETWEEN WORDS
C
C ROUTINES: NONE
C
C NOTES:    IF THERE IS NO Nfirst WORD OR NO WORDS ARE FOUND
C           (I.E. INPUT STRING IS BLANK) THEN IWORDS=0
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/37
C          JET EXT. 5023
C
C DATE:    20/05/93
C
C-----------------------------------------------------------------------
      CHARACTER*(*)       CDELIM,      CTEXT
      INTEGER             IFIRST(IWORDS),           ILAST(IWORDS)
      INTEGER             IWORDS,      NFIRST,      NWORDS
```

9.200 xxwstr: Subroutine xxwstr from library adaslib

```
C
      subroutine xxwstr(iunit,string)


C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXWSTR *********************
C
C  PURPOSE: Writes a string with format (a) to iunit with no
C           trailing blanks.
C
C  CALLING PROGRAM: GENERAL USE.
C
C  INPUT    : (C*(*)) STRING = STRING
C
C  INPUT    : (I*4)   IUNIT  = OUTPUT UNIT
C
C  ROUTINES :
C            ROUTINE    SOURCE    BRIEF DESCRIPTION
C            ------------------------------------------------------------
C            XXSLEN     ADAS      GETS NON-BLANK LENGTH OF A STRING
C
C  AUTHOR   : Martin O'Mullane,
C             K1/1/43,
C             JET
C
C  VERSION  : 1.1                          DATE: 17/03/1999
C  MODIFIED : Martin O'Mullane
C             First version.
C
C-----------------------------------------------------------------------
      CHARACTER*(*)       STRING
      INTEGER             IUNIT
```

9.201 xxycf2: Subroutine xxycf2 from library adaslib

```
      SUBROUTINE XXYCF2(FCN, M, N, X, FVEC, FJAC, B, C)
C-----------------------------------------------------------------------
C
C ROUTINE: XXYCF2
C
C PURPOSE: CALCULATES AN ESTIMATE OF THE ELEMENTS OF THE VARIANCE-COVARIANCE
C          MATRIX OF THE ESTIMATED REGRESSION COEFFICIENTS FOR A NON-LINEAR
C          LEAST-SQUARES PROBLEM.
C
C          REPLACES NAG ROUTINE E04YCF WHEN CALLED AFTER E04GCF OR ANOTHER
C          ROUTINE THAT USES FIRST DERIVATIVES.
C***********************************************************************
C          THIS ROUTINE CAN ONLY BE USED WHEN FIRST DERIVATIVES ARE AVAILABLE.
C***********************************************************************
C          SEE THE DOCUMENTATION OF E04YCF AND THE E04 SERIES FOR MORE
C          INFORMATION ON THE MATHEMATICS OF CALCULATING THE VARIANCE-
C          COVARIANCE MATRIX.
C
C CALLING PROGRAM: GENERAL USE
C
C INPUT:
C    SUBROUTINE FCN - CALCULATES THE FUNCTIONS AND THEIR DERIVATIVES.
C                     IF ONE IS DIRECTLY REPLACING THE NAG ROUTINE THEN
C                     THIS ARGUMENT WILL BE LSFUN2 IN THE CALLING
C                     PROGRAM, AND LSFUN2 WILL HAVE TO BE MODIFIED TO
C                     ACCOMMODATE THE IFLAG VARIABLE. SEE BELOW FOR DETAILS.
C    (I*4)  M       - NUMBER OF FUNCTIONS
C    (I*4)  N       - NUMBER OF VARIABLES, N<=M
C    (R*8)  X       - VECTOR OF THE ESTIMATED SOLUTION
C    (R*8)  FVEC()  - THE FUNCTION EVALUATED AT X
C    (R*8)  FJAC()  - A WORK MATRIX OF DIMENSION (M,N)
C    (R*8)  B()     - A WORK VECTOR OF DIMENSION N
C
C OUTPUT:
C     (R*8)  C(,)     - THE VARIANCE-COVARIANCE MATRIX. DIMENSIONS NxN.
C
C ROUTINES:
C-----------------------------------------------------------------------
C     NAME     SOURCE   PURPOSE
C-----------------------------------------------------------------------
C     XXMINV   ADAS     INVERTS A SQUARE MATRIX
C     FCN      USER     CALCULATES FUNCTIONS AND THEIR FIRST DERIVATIVES
C {
C    SUBROUTINE FCN(M,N,X,FVEC,FJAC,LDFJAC,IFLAG)
C    INTEGER M,N,IFLAG, LDFJAC
C    REAL*8 X(N), FVEC(M), FJAC(LDFJAC,N)
C  INPUT:
C    M,N    - AS ABOVE
C    LDFJAC - FIRST DIMENSION OF FJAC
C    X      - VECTOR AT WHICH THE FUNCTION IS TO BE EVALUATED
C    IFLAG  - 1=> CALCULATE FVEC BUT DON'T CHANGE FJAC
C           - 2=> CALCULATE FJAC BUT DON'T CHANGE FVEC
C  OUTPUT:
C    FVEC   - FUNCTION EVALUATED AT X OR NOT CHANGED
C    FJAC   - JACOBIAN OF FUNCTION EVALUATED AT X OR NOT CHANGED
C }
C-----------------------------------------------------------------------
C
C AUTHOR: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC.)
```

```
C
C DATE:   31-05-96
C
C VERSION 1.1                              DATE: 31-05-96
C MODIFIED: WILLIAM OSBORN
C           - FIRST VERSION
C
C VERSION 1.2                              DATE: 16-05-07
C MODIFIED: ALLAN WHITEFORD
C           - Updated comments as part of subroutine documentation
C             procedure.
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      INTEGER             M,           N
      REAL*8              B(N),        C(N,N),      FJAC(M,N),   FVEC(M)
      REAL*8              X(N)
```

10 Practical guidance and examples

10.1 Calling subroutines from Fortran

Since almost all ADAS subroutines are written in Fortran, calling them from Fortran tends to be much more straightforward than using C or C++. A typical compilation command to include the (e.g.) adas3xx library might be:

```
f77 myfile.f -o myfile.x -L/home/adas/lib -ladas3xx
```

the location of `/home/adas/lib` is installation/site specific and should be determined locally. It is frequently the case that the subroutine libraries rely on other libraries, particularly the generic `adaslib` library which underpins many of the series specific routines so it is often necessary to include more than one library. Examples of Fortran programs which use ADAS can be found in sections 10.3 and 10.4. Suggested compilation commands are given in the documentation of these programs.

10.2 Calling subroutines from C and C++

Calling ADAS subroutines from C is more problematic than Fortran. A particular problem is that a C compiler will not, by default, pass on underlying Fortran system (i.e. non-ADAS) libraries to the linker stage. The compilation options can be quite involved. This is an issue generic to linking Fortran & C and is not specific to ADAS. Guidance can be sought from either local system support or ADAS. It is also possible to compile a short dummy Fortran program and use the `--verbose` (or similar) compiler option to see what is passed to the linker.

However, for modern GNU/Linux systems with gcc, it is possible to compile C programs with the "g77" command, this has the side effect of also passing Fortran specific options to the linker, thus:

```
g77 myfile.c -L/home/adas/lib -ladaslib -o myfile.x
```

can often be effective (although note that gfortran does not have this useful side effect if no fortran codes are specified at the command line). A typical compilation command only using a C compiler (in this case gcc version 3) could look like:

```
gcc myfile.c -o myfile.x -L/home/adas/lib -ladaslib -lg2c
```

but other libraries in addition to "g2c" may be necessary. On some systems it may also be necessary to explicitly use version 3 of gcc (the last version that provided backward compatibility with g77), this is typically installed as the command 'gcc32', 'gcc34' etc. (note that 'gcc32' is not related to 32 bit vs 64 bit compilation). If the ADAS libraries have been compiled with gfortran then a typical compilation command would look like:

```
gcc myfile.c -o myfile.x -L/home/adas/lib -ladaslib -lgfortran
```

but, as above, other libraries may be necessary[1]

A great deal of confusion can be caused by the appending of trailing underscores and trailing double underscores on to Fortran routines. All ADAS library routines are compiled with the '-fno-second-underscore' flag set. This means that a subroutine called 'FOO' in Fortran becomes a C void function called 'foo_'. However, care must be taken when linking routines with each other. Error messages will be given at the linker stage unless you are building shared libraries (as opposed to an executable program) in which case error messages are delayed until runtime.

10.2.1 Variable references, strings and Fortran I/O units

All Fortran subroutines take in references to variables, even to scalar variables. This is in contrast to C which passes everything by value (in this context, the 'value' of an array is defined as a pointer to the start of the array — the semantics of this statement are unimportant). This means that given a Fortran subroutine 'FOO' defined thus:

```
SUBROUTINE FOO(A)
INTEGER*4 A

PRINT *,A
```

[1] ADAS is typically compiled using g77 (i.e. gcc version 3) but gfortran is used if g77 is not present on a given system. A request can be made for libraries compiled using a different compiler.

```
        END
```

which would have a C prototype of:

```
extern foo_(int *a);
```

it is **NOT** acceptable to write a C program thus:

```
extern foo_(int *a);
int main(void)
{
         foo_(15);
}
```

instead, one needs to write:

```
extern foo_(int *a);
int main(void)
{
         int x=15;
         foo_(&x);
}
```

this is so that Fortran receives a memory address to an integer (note that it's also possible to solve the issue in Fortran using things like %VAL — this is not discussed here since all ADAS Fortran routines are written conventionally and expect memory addresses).

Passing arrays is the same but the syntax is slightly different. Again, given a subroutine 'FOO':

```
        SUBROUTINE FOO(A)
        INTEGER*4 A(10)

        PRINT *,A(5)

        END
```

then the correct calling sequence would be:

```
extern foo_(int *a);
int main(void)
{
         int z[10];
         z[4]=15;
         foo_(z);
}
```

note that C counts from 0 and Fortran counts from 1 so that z[4] in C translates to A(5) in Fortran. Also note that our explicit memory addressing (i.e. with &) of the integer variable is no longer necessary since an array is already a memory address). For a 2D array our Fortran routine becomes:

```
        SUBROUTINE FOO(A)
        INTEGER*4 A(10,20)

        PRINT *,A(5,8)

        END
```

and our C routine is now:

```
extern foo_(int *a);

int main(void)
{
        int z[20][10];
        z[7][4]=15;
        foo_(*z);
}
```

note specifically that we now dereference 'z' once (by writing '*z') since we want to find the memory address of the first element. Also worthy of note is that the array indices swap between C and Fortran, i.e. [7][4] in C becomes (5,8) in Fortran. For a 3D array, as one might expect we end up with:

```
      SUBROUTINE FOO(A)
      INTEGER*4 A(10,20,30)

      PRINT *,A(5,8,19)

      END
```

and:

```
extern foo_(int *a);

int main(void)
{
        int z[30][20][10];
        z[18][7][4]=15;
        foo_(**z);
}
```

note the double dereferencing of 'z' (i.e. '**z'). Higher dimensioned arrays are as one would expect.

Passing strings from C to Fortran is problematic and highly compiler dependent. Fortran subroutines pass the length of strings to each other as hidden parameters; if working only in the Fortran domain then the programmer never needs to worry about these parameters but if calling a Fortran routine from C you need to pass these parameters. It is implementation specific where the lengths are passed but most compilers append them at the very end of the argument list; if there is more than one string being passed then the order of the lengths are the same as the order of the original strings. As an example, a Fortran routine which accepts variable length strings such as:

```
        subroutine foo(a,b,c,d)

        character*(*)   a
        integer*4       b
        character*(*)   c
        integer*4       d

        print *,a,b
        print *,c,d

        end
```

has a prototype of:

```
extern foo_(char *a, int *b, char *c, int *d, int len_a, int len_c);
```

and should be called thus:

```
extern foo_(char *a, int *b, char *c, int *d, int len_a, int len_c);

int main(void)
{
        char a[20];
        int b;
        char c[20];
        int d;

        strcpy(a,"Example String A");
        b=1;
        strcpy(c,"Example String C");
        d=2;

        foo_(a,&b,c,&d,strlen(a),strlen(c));
}
```

note that we don't say '&a' or '&c' since 'a' and 'c' are already a memory addresses. Note also that len_a and len_c are passed by value rather than by reference.

If the Fortran routine has fixed length strings such as:

```
        subroutine foo(a,b,c,d)

        character*40    a
        integer*4       b
        character*40    c
        integer*4       d

        print *,a,b
        print *,c,d

        end
```

then it is necessary to fill the strings with NULL characters (i.e. '\0') in C before passing them to the Fortran. Note that C by default only puts a NULL character at the end of the string text, but the rest of the memory is essentially random, which is insufficient for Fortran. A C routine would look like:

```
extern foo_(char *a, int *b, char *c, int *d, int len_a, int len_c);

int main(void)
{
        char a[41];
        int b;
        char c[41];
        int d;
        int i;

        for (i=0;i<=40;i++)
        {
                a[i]='\0';
                c[i]='\0';
        }

        strcpy(a,"Example String A");
```

```
        b=1;
        strcpy(c,"Example String C");
        d=2;

        foo_(a,&b,c,&d,40,40);

}
```

most Fortran compilers will typically ignore the lengths being passed from C if the lengths are explictly defined in the Fortran. Note also that a C string has to be one character longer than a Fortran string because C strings require a trailing '\0' when at fully capacity whereas Fortran strings have an explicit length so do not. It's possible to code the C such that this isn't necessary but then there is a chance that your strings will behave poorly with functions such as printf, strcpy etc. Note that it is often possible to 'get away' with not padding out the strings with NULLs, particularly when only using the strings as filenames. This is not recommended.

Things are slightly more problematic for string arrays, given a Fortran subroutine:

```
        subroutine foo(a,b)

        character*40    a(15)
        character*20    b(10)

        print *,a(1)
        print *,b(4)

        end
```

we would use a C program such as:

```
extern foo_(char *a, char *b,int len_a, int len_b);

int main(void)
{
        char a[15][40];
        char b[10][20];
        int i,j;

        for (i=0;i<15;i++)
        {
                for (j=0;j<40;j++)
                {
                        a[i][j]='\0';
                }
        }

        for (i=0;i<10;i++)
        {
                for (j=0;j<20;j++)
                {
                        b[i][j]='\0';
                }
        }

        strcpy(a[0],"Example String A");
        strcpy(b[3],"Example String B");
```

```
        foo_(*a,*b,40,20);
}
```

there are a number of things worthy of note. Firstly, a and b are now (in the C sense) 2D arrays so we need to pass in '*a' as we did before with 2D integer arrays. There are clearly more efficient (and less verbose) ways of clearing the memory contents of 'a' and 'b' but the code was written as above for reasons of clarity.

The most subtle issue is now the C strings have to have the same number of characters as the Fortran strings (previously we dimensioned our C strings with one extra character). Due to the assumptions that Fortran makes about memory it's no longer possible to have this extra padding character for the sake of C. The consequences of this are not severe: even though we have, say, 40 character strings in Fortran we should only every treat them with an effective length of 39. This means that in C they can still end with a '\0'. If we use the whole 40 characters in Fortran then the strings will not behave well when passed to C functions such as printf, strcpy etc..

Many ADAS routines, especially the `xxdata` series of routines take Fortran unit numbers as an input rather than filenames. They expect the file to be opened by the calling subroutine, this is present for a number of reasons but can pose a problem since a C filehandle is not the same as a Fortran unit number. To get around this we suggest calling the ADAS library routine XXOPEN to obtain open a file on a unit number. The routine can be called:

```
        long lexist;
        int iunit=10;
        char dsnin[80];
        strcpy(dsnin,"/home/adas/adas/path/to/datafile.dat");
        xxopen_ (&iunit, dsnin, &lexist, strlen(dsnin));
```

which will open the above file on Fortran unit 10, the unit variable can then be passed in to the required routine. An example of using XXOPEN in a practical application is given in section 10.6. Note also that the ADAS routine XXOPCS can open a file and convert it to all lower case or all upper case as it is read.

10.2.2 Header files

For ease of use, many ADAS Fortran subroutines have C-style prototypes generated and these can normally be found in `/home/adas/include` but this path is site specific. A header file for each ADAS series exists and are called, e.g. adas3xx.h for ADAS series 3 subroutines and adaslib.h for generic ADAS library subroutines. These can be included at the top of C programs and the path to the include files should be specified with the '-I' command line. For example, a complete program to open a file on unit 10 (albeit then do nothing with it) would look like:

```
#include <string.h>
#include "adaslib.h"

int main(void)
{
        long lexist;
        int iunit=10;
        char dsnin[80];
        strcpy(dsnin,"/home/adas/adas/path/to/datafile.dat");
        xxopen_ (&iunit, dsnin, &lexist, strlen(dsnin));
}
```

and would be compiled with one of:

```
g77 -I$ADASTOP/include small.c -L$ADASTOP/lib -ladaslib
gcc -I$ADASTOP/include small.c -L$ADASTOP/lib -ladaslib -lg2c
gcc -I$ADASTOP/include small.c -L$ADASTOP/lib -ladaslib -lgfortran
```

where note that in the first example we're using the g77 command to compile C. You must use the same compiler version as ADAS was compiled with — see section 10.2 for more details on compiler issues with specific reference to g77 versus gfortran and the associated issue of gcc version 3 versus version 4 and the occasional neccesity to use a command like 'gcc32'.

Section 10.5 gives an example C program which calculates bremmstrahlung (analogous to the Fortran example given in 10.3) and section 10.7 gives an example of using C to extract beam stopping coefficients from ADF21 files.

10.2.3 Specific C++ issues and usage

The main difference between calling an ADAS code from C++ as opposed to C is that a prototype which would have been:

```
extern void foo_(int *a, char *b, float *c );
```

in C must become:

```
extern "C" { void foo_(int *a, char *b, float *c ); };
```

in C++. All of the ADAS header files automatically detect what language they are being compiled into (via looking for the definition of '__cplusplus') and prototypes are defined accordingly.

All of the information on linking with C given in the previous section is still valid but note that the C++ string type needs to converted to a character array before going in to Fortran. Given a Fortran routine such as:

```
        subroutine foo(a,b,c,d)

        character*30    a
        real*4          b(10,20)
        character*40    c(10)
        integer         d

        print *,a
        print *,b(10,18)
        print *,c(3)
        print *,d

        end
```

and a simple C++ program such as:

```
using namespace std;
#include <string>

int main(void)
{
        string a;
        float  b[20][10];
        string c[10];
        int    d;

        a="Example string in scalar";
        b[17][9]=123.0;
        c[2]="Example string in array";
        d=3;
}
```

then in order to call the Fortran FOO routine the C++ would need to look like:

```
extern "C" { void foo_(char *a,float *b       ,char *c,
                       int  *d,int   strlen_a ,int strlen_c);
           };
using namespace std;
#include <string>

int main(void)
{
        string a;
        float  b[20][10];
        string c[10];
        int    d;

        a="Example string in scalar";
        b[17][9]=123.0;
        c[2]="Example string in array";
        d=3;

        char a_temp[30];
        char c_temp[10][40];
        int i,j;

        for (i=0;i<30;i++) a_temp[i]='\0';
        if (!a.empty())
        {
                strcpy(a_temp,a.c_str());
        }

        for (i=0;i<10;i++)
        {
                for (j=0;j<40;j++)
                {
                        c_temp[i][j]='\0';
                }
                if (!c[i].empty())
                {
                        strcpy(c_temp[i],c[i].c_str());
                }
        }

        foo_(a_temp,*b,*c_temp,&d,30,40);

}
```

where the example given above is coded quite verbosely. Note that this shows an example of passing a string array, scalar string, scalar inteer and 2D floating point array in one example. Readers are encouraged to read the previous section on calling routines from C for a discussion of each of these separate issues.

In the routine above the '`c_str()`' method the C++ string class has been used to return an input suitable for strcpy but that it is necessary before invoking this method to check that the string isn't empty.

Compilation issues also arise when using C++. For straightforward C and Fortran linking using GNU tools it was suggested to simply invoke a Fortran compiler to take care of extra linking. However, using a Fortran compiler does not link in extra C++ libraries (conversely, using a C++ compiler doesn't link in any Fortran libraries). Suggested generic commands are:

```
g++ -I/home/adas/include myfile.cpp -L/home/adas/lib -ladaslib -lgfortran
```

for gcc 4 compiler (i.e. a gfortran implementation) and:

```
g++ -I/home/adas/include myfile.cpp -L/home/adas/lib -ladaslib -lg2c
```

for a gcc 3 compiler (i.e. a g77 implementation). Note that on modern systems a backward compatible gcc version 3 C++ compiler will typically be a command like 'g++32', 'g++34' or similar (see section 10.2).

Note that the C code given in section 10.7 which calculates a beam stopping coefficient is also perfectly valid C++ code. However, also see the next section for details on using C++ in a more idiomatic (i.e. object-based) way.

10.2.4 Class wrappers for C++ usage

There are a limited[2] number of class wrappers available for ADAS subroutines. The headers are arranged on a per-subroutine basis instead of a by series basis. The header files have an extension of `.hpp`. The definitive list of subroutines available with C++ wrappers can be found by looking for `.hpp` files in the `/home/adas/include` directory.

Note that it's possible to call every ADAS subroutine from C++ (see the previous section) and we are only presently discussing class-based solutions to calling ADAS subroutines.

Documentation for the classes appears in the header files but the relevant Fortran documentation should also be consulted for a fuller description of the workings of the particular subroutine.

All of the C++ classes are inside the 'adas' namespace to avoid symbol clashes with any other routines you may have, this means that to declare an ADAS object one would have something like:

```
#include "cxbms.hpp"     // Include header file
adas::cxbms myobject;    // Declare object
```

Compilation against these classes requires linking in the 'adascpp' library so generic compiler commands when using class wrappers become:

```
g++ -I/home/adas/include myfile.cpp -o program.x  \
         -L/home/adas/lib -ladascpp -ladaslib -lgfortran
```

for gcc 4 compiler (i.e. a gfortran implementation) and:

```
g++ -I/home/adas/include myfile.cpp -o program.x \
         -L/home/adas/lib -ladascpp -ladaslib -lg2c
```

for a gcc 3 compiler (i.e. a g77 implementation). Note that on modern systems a backward compatible gcc version 3 C++ compiler will typically be a command like 'g++32', 'g++34' or similar (see section 10.2). Also, it may be necessary to link in other ADAS libraries depending on what the C++ class is wrapping.

An example of using a class interface to the same example of beam stopping as exemplified in section 10.7 is given in 10.8.

[2] These are added on a per-request basis.

10.3 Fortran example: Bremsstrahlung emission

```
       PROGRAM BREM
C-----------------------------------------------------------------------
C
C  ******************* FORTRAN77 PROGRAM: BREM ************************
C
C  PURPOSE:   TO EVALUATE BREMSSTRAHLUNG EMISSION USING ADAS AT A
C             SPECIFIED TEMPERATURE AND DENSITY. ORIGINALLY WRITTEN AS
C             AN EXAMPLE PROGRAM FOR RITU DEY AT THE IPR
C
C  ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          CONTINUO   ADAS      FOR AN GIVEN WAVELENGTH GENERATE
C                               RADIATIVE RECONBINATION AND
C                                BREMSSTRAHLUNG EMISSIVITY.
C
C  NOTES :    CONTAINS VERBOSE COMMENTS TO ILLUSTRATE HOW ADAS
C             MAY BE USED FROM STAND ALONE FORTRAN PROGRAMS.
C
C             THIS PROGRAM SHOULD BE LINKED TO BOTH THE ADAS3XX
C             AND ADASLIB LIBRARIES, COMPILATION WILL RESEMBLE:
C               f77 brem.for -L/home/adas/lib \
C                       -ladas3xx -ladaslib -o brem.x
C
C
C  AUTHOR:    ALLAN WHITEFORD
C             UNIVERSITY OF STRATHCLYDE
C
C  DATE:      30/03/06
C-----------------------------------------------------------------------

C-----------------------------------------------------------------------
C      DEFINE APPOPRIATE VARIABLES
C-----------------------------------------------------------------------
       IMPLICIT  NONE
C-----------------------------------------------------------------------
       INTEGER*4 IZ0  , IZ1
       REAL*8    WAVE , TEV , CONTFF , CONTIN
C-----------------------------------------------------------------------

C-----------------------------------------------------------------------
C      SPECIFY WE'RE DEALING WITH FULLY STRIPPED CARBON
C-----------------------------------------------------------------------
       IZ0=6
       IZ1=7

C-----------------------------------------------------------------------
C      SPECIFY A TEMPERATURE OF 3keV and a WAVELENGTH OF 6000A
C-----------------------------------------------------------------------
       TEV=3000
       WAVE=6000

C-----------------------------------------------------------------------
C      USE ADAS TO CALCULATE EMISSION
C-----------------------------------------------------------------------
       CALL CONTINUO(WAVE   , TEV    , IZ0  , IZ1  ,
     &               CONTFF , CONTIN
     &                                          )


C-----------------------------------------------------------------------
C      PRINT OFF FREE-FREE EMISSIVITY (Ph cm3 s-1 A-1)
C-----------------------------------------------------------------------
       PRINT *,CONTFF

C-----------------------------------------------------------------------
C      PRINT OFF TOTAL EMISSIVITY (Ph cm3 s-1 A-1)
C-----------------------------------------------------------------------
       PRINT *,CONTIN

       END
```

10.4 Fortran example: Ionisation balance for carbon

```
       PROGRAM CARBON
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 PROGRAM: CARBON ************************
C
C  PURPOSE:   TO EVALUATE EQUILIBRIUM IONISATION BALANCE FOR CARBON AT
C             A RANGE OF FIXED TEMPERATURE AND DENSITY PAIRS. WRITTEN
C             AS AN EXAMPLE PROGRAM FOR PAVEL GONCHAROV AT NIFS/LHD AS
C             A DEMONSTRATION OF HOW TO CALL ADAS LIBRARIES FROM A
C             FORTRAN PROGRAM.
C
C  ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          D5DATA     ADAS      FETCHES DATA FROM ADF11 MASTER FILES
C          D5MPOP     ADAS      PARTITIONED TRI-DIAG MATRIX INVERSION
C
C  NOTES :    CONTAINS VERBOSE COMMENTS TO ILLUSTRATE HOW ADAS
C             MAY BE USED FROM STAND ALONE FORTRAN PROGRAMS.
C
C             THIS PROGRAM SHOULD BE LINKED TO BOTH THE ADAS4XX
C             AND ADASLIB LIBRARIES, COMPILATION WILL RESEMBLE:
C               f77 carbon.for -L/home/adas/lib \
C                       -ladas4xx -ladaslib -o carbon.x
C
C
C  AUTHOR:    ALLAN WHITEFORD
C             UNIVERSITY OF STRATHCLYDE
C
C  DATE:      01/02/07
C-----------------------------------------------------------------------

C-----------------------------------------------------------------------
C      DEFINE APPOPRIATE VARIABLES
C-----------------------------------------------------------------------
       IMPLICIT  NONE
C-----------------------------------------------------------------------
       INTEGER   NTDIM   , ITDIMD , IZDIMD , IMDIMD ,
     &           NDFILE  , IPDIMD , ISDIMD , NDONE
C-----------------------------------------------------------------------
       PARAMETER ( NTDIM  = 30 , ITDIMD = 51 , IZDIMD = 83 )
       PARAMETER ( IPDIMD = 5  , ISDIMD = 83 , IMDIMD = ISDIMD+1 )
       PARAMETER ( NDONE = 1   , NDFILE = 6  )
C-----------------------------------------------------------------------
       LOGICAL LPART , LSOLVE
C-----------------------------------------------------------------------
       LOGICAL LSELA(8)  , LEXSA(8)   , LDEFA(8)
       LOGICAL LACDA(IZDIMD,IPDIMD,IPDIMD),
     &         LSCDA(IZDIMD,IPDIMD,IPDIMD),
     &         LCCDA(IZDIMD,IPDIMD,IPDIMD),
     &         LQCDA(IZDIMD,IPDIMD,IPDIMD),
     &         LXCDA(IZDIMD,IPDIMD,IPDIMD),
     &         LPRBA(IZDIMD,IPDIMD),
     &         LPRCA(IZDIMD,IPDIMD),
     &         LPLTA(IZDIMD,IPDIMD)
C-----------------------------------------------------------------------
       CHARACTER DSFLLA(8)*120
C-----------------------------------------------------------------------
       INTEGER   IZ0
       INTEGER   I
       INTEGER   ITMAX
       INTEGER   IZL     , IZH     ,
     &           IT      ,
     &           NSTAGE  , NMSUM
C-----------------------------------------------------------------------
       INTEGER   NPRT(IZDIMD)  ,  NPRTR(IZDIMD)
C-----------------------------------------------------------------------
       REAL*8    DNS        , DNSH
C-----------------------------------------------------------------------
       REAL*8    TEV(NTDIM) , DTEV(NTDIM) ,
     &           DENS(NTDIM), DDENS(NTDIM), DENSH(NTDIM),
     &           FPABUN(NTDIM,IMDIMD)
       REAL*8    DTEVD(ITDIMD) , DDENSD(ITDIMD) , ZDATA(ISDIMD),
     &           DRCOFD(ISDIMD,ITDIMD,ITDIMD),
     &           DRCOFI(NTDIM)
```

```
       REAL*8    ACDA(NTDIM,IZDIMD,IPDIMD,IPDIMD),
     &           SCDA(NTDIM,IZDIMD,IPDIMD,IPDIMD),
     &           CCDA(NTDIM,IZDIMD,IPDIMD,IPDIMD),
     &           QCDA(NTDIM,IZDIMD,IPDIMD,IPDIMD),
     &           XCDA(NTDIM,IZDIMD,IPDIMD,IPDIMD),
     &           PRBA(NTDIM,IZDIMD,IPDIMD),
     &           PRCA(NTDIM,IZDIMD,IPDIMD),
     &           PLTA(NTDIM,IZDIMD,IPDIMD)
       REAL*8    POPF(IMDIMD)
       REAL*8    CFREC(IPDIMD,IPDIMD,IZDIMD),
     &           CFION(IPDIMD,IPDIMD,IZDIMD),
     &           CFMET(IPDIMD,IPDIMD,IZDIMD)
       REAL*8    CPOPN(IPDIMD,IPDIMD,IZDIMD+1) ,
     &           CPOPND(IPDIMD,IPDIMD,IZDIMD+1),
     &           CPOPNZ(IPDIMD,IPDIMD,IZDIMD+1)
       REAL*8    POPNMO(IPDIMD,NDONE,IZDIMD+1),
     &           POPNPO(IPDIMD,NDONE,IZDIMD+1),
     &           POPN(IPDIMD,NDONE,IZDIMD+1)
       REAL*8    RDUM(IPDIMD),RHS(2*IPDIMD-1),
     &           SOLVE(2*IPDIMD-1,2*IPDIMD-1)
       REAL*8    XTEMP(IPDIMD,IPDIMD),YTEMP(IPDIMD,IPDIMD)
       REAL*8    YTEM(IZDIMD)

C-----------------------------------------------------------------------
C      SET LOCATION OF RECOMBINATION (ACD) AND IONISATION (SCD) FILES
C-----------------------------------------------------------------------
       DSFLLA(1)='/home/adas/adas/adf11/acd96/acd96_c.dat'
       DSFLLA(2)='/home/adas/adf11/scd96/scd96_c.dat'
       LSELA(1)=.TRUE.
       LSELA(2)=.TRUE.
       LEXSA(1)=.TRUE.
       LEXSA(2)=.TRUE.

C-----------------------------------------------------------------------
C      SPECIFY THAT FILES ARE NOT METASTABLE
C-----------------------------------------------------------------------
       LPART=.FALSE.
       NSTAGE=7
       NMSUM=7
       NPRT(1)=1
       NPRT(2)=1
       NPRT(3)=1
       NPRT(4)=1
       NPRT(5)=1
       NPRT(6)=1
       NPRT(7)=1

C-----------------------------------------------------------------------
C      INDICATE THAT FILES CONTAIN COMPLETE INFORMATION
C-----------------------------------------------------------------------
       IZ0=6
       IZL=1
       IZH=7

C-----------------------------------------------------------------------
C      SET 13 TEMPERATURE AND DENSITY PAIRS TO COMPUTE IONISATION
C      BALANCE AT. IN THIS EXAMPLE WE COVER A WIDE RANGE AND ASSUME
C      CONSTANT DENSITY. HERE YOU COULD USE THE DATA FROM THOMSON
C      SCATTERING MEASUREMENTS OR SIMILAR
C-----------------------------------------------------------------------
       ITMAX=13

       DENS(1)=1d14
       DENS(2)=1d14
       DENS(3)=1d14
       DENS(4)=1d14
       DENS(5)=1d14
       DENS(6)=1d14
       DENS(7)=1d14
       DENS(8)=1d14
       DENS(9)=1d14
       DENS(10)=1d14
       DENS(11)=1d14
       DENS(12)=1d14
       DENS(13)=1d14
```

```
       TEV(1)= .1d0
       TEV(2)= .2d0
       TEV(3)= .5d0
       TEV(4)= 1d0
       TEV(5)= 2d0
       TEV(6)= 5d0
       TEV(7)= 10d0
       TEV(8)= 20d0
       TEV(9)= 50d0
       TEV(10)= 100d0
       TEV(11)= 200d0
       TEV(12)= 500d0
       TEV(13)= 1000d0

C-----------------------------------------------------------------------
C      SUBROUTINE ALSO REQUIRE THAT LOG10 OF THE ABOVE ARE PASSED
C-----------------------------------------------------------------------
       DO 10 I=1,ITMAX
10             DDENS(I)= DLOG10(DENS(I))
       DO 11 I=1,ITMAX
11             DTEV(I)= DLOG10(TEV(I))

C-----------------------------------------------------------------------
C      CALL ADAS ROUTINE TO READ THE ACD AND SCD FILES, THIS ROUTINE
C      ALSO INTERPOLATES ON TO THE DESIRED TEMPERATURE AND DENSITIES
C-----------------------------------------------------------------------
       CALL D5DATA( DSFLLA , LSELA  , LEXSA  , LDEFA  , LPART  ,
     &              IZ0    , IZL    , IZH    , NPRT   ,
     &              NTDIM  , ITMAX  ,
     &              ISDIMD , IZDIMD , ITDIMD , IPDIMD , NPRTR  ,
     &              DTEV   , DDENS  ,
     &              DTEVD  , DDENSD , DRCOFD , ZDATA  , DRCOFI ,
     &              ACDA   , LACDA  ,
     &              SCDA   , LSCDA  ,
     &              CCDA   , LCCDA  ,
     &              PRBA   , LPRBA  ,
     &              PRCA   , LPRCA  ,
     &              QCDA   , LQCDA  ,
     &              XCDA   , LXCDA  ,
     &              PLTA   , LPLTA  )


C-----------------------------------------------------------------------
C      LOOP OVER TEMERATURE AND DENSITY PAIRS
C-----------------------------------------------------------------------
       DO 12 IT=1,ITMAX
           DNS  = DENS(IT)
           DNSH = DENSH(IT)

C-----------------------------------------------------------------------
C          CALL ADAS ROUTINE TO CALCULATE THE FRACTIONAL ABUNDANCIES
C          OF THE DIFFERENT STAGES
C-----------------------------------------------------------------------

           CALL D5MPOP( NTDIM , IZDIMD, IPDIMD,
     &                  NSTAGE, ITMAX , NPRT  , NMSUM ,
     &                  ACDA  , SCDA  , CCDA  , QCDA  , XCDA  ,
     &                  DENS  , DENSH ,
     &                  IT    ,
     &                  CFREC , CFION , CFMET ,
     &                  POPN  , POPNMO, POPNPO,
     &                  CPOPN , CPOPND, CPOPNZ,
     &                  POPF  ,
     &                  XTEMP , YTEMP , YTEM  ,
     &                  RHS   , RDUM  , SOLVE , LSOLVE        )

C-----------------------------------------------------------------------
C          STORE THE RESULTS IN FPABUN, SET MINIMUM TO 1D-74 TO PROTECT
C          FOR UNDERFLOWS LATER IN THE PROGRAM
C-----------------------------------------------------------------------
           DO 13 I=1,NMSUM
   13          FPABUN(IT,I) = MAX(POPF(I),1D-74)

C-----------------------------------------------------------------------
C      END LOOP OVER TEMERATURE AND DENSITY PAIRS
C-----------------------------------------------------------------------
```

```
  12   CONTINUE

C-----------------------------------------------------------------------
C      WRITE THE OUTPUT TO THE SCREEN, IN A REAL EXAMPLE OF COURSE
C      YOU WOULD THEN USE THE CALCULATED DATA
C-----------------------------------------------------------------------
       WRITE(6,1000) (I-1,I=1,7)
       DO 14 IT=1,ITMAX
  14       WRITE(6,1001) TEV(IT),(FPABUN(IT,I),I=1,7)

1000   FORMAT('#',2X,'T / eV',5X,7('C+',I1,7X))
1001   FORMAT(8(1PE10.2))

C-----------------------------------------------------------------------
C      END OF EXAMPLE
C-----------------------------------------------------------------------
       END
```

10.5 C example: Bremsstrahlung emission

```
/*
 *      ********************* C Program: brem *********************
 *
 * Purpose:  To evaluate bremsstrahlung emission using adas at a
 *           specified temperature and density. originally written as
 *           an example program (in Fortran) for Ritu Dey at the IPR.
 *
 * ROUTINES:
 *         ROUTINE    SOURCE    BRIEF DESCRIPTION
 *         ------------------------------------------------------------
 *         continuo   ADAS      For a given wavelength generate
 *                              radiative reconbination and
 *                              bremsstrahlung emissivity.
 *
 * NOTES :    Contains verbose comments to illustrate how ADAS
 *            may be used from stand alone C programs.
 *
 *            This program should be linked to both the adas3xx
 *            and adaslib libraries, compilation will resemble:
 *              g77 -I/home/adas/include brem.c -L/home/adas/lib \
 *                      -ladas3xx -ladaslib -o brem.x
 *            (Note that g77 is used for compilation even though this
 *             program is written in C)
 *
 * AUTHOR:    Allan Whiteford
 *            University of Strathclyde
 *
 * DATE:      23/03/07
 *
 */

/*****************************************************************************
 Bremsstrahlung routine is in adaslib library so include appropriate header
*****************************************************************************/
#include "adaslib.h"

int main(int argc, char *argv)
{

/*****************************************************************************
        Declare appopriate variables
*****************************************************************************/
        int iz0, iz1;
        double wave,tev,contff,contin;

/*****************************************************************************
        Specify we're dealing with fully stripped carbon
*****************************************************************************/
        iz0=6;
        iz1=7;

/*****************************************************************************
        Specify a temperature of 3keV and a wavelength of 6000A
*****************************************************************************/
        wave=6000;
        tev=3000;

/*****************************************************************************
        Use ADAS to calculate emission
*****************************************************************************/
        continuo_(&wave   , &tev    , &iz0  , &iz1  ,&contff , &contin);

/*****************************************************************************
        Print off free-free emissivity (Ph cm3 s-1 A-1)
*****************************************************************************/
        printf("%e\n",contff);

/*****************************************************************************
        Print off total emissivity (Ph cm3 s-1 A-1)
*****************************************************************************/
        printf("%e\n",contin);

        return 0;
}
```

10.6 C example: Reading an ADF21 file

```
/*
 *      ********************* C Program: adf21 ********************
 *
 * Purpose:  To read an ADF21 file using ADAS. Originally written as
 *           an example program for Masaki Osakabe at LHD/NIFS.
 *
 * ROUTINES:
 *         ROUTINE    SOURCE    BRIEF DESCRIPTION
 *         ------------------------------------------------------------
 *         xxopen     ADAS      Opens a file on a Fortran unit number
 *         xxdata_21  ADAS      Reads an ADF21 file
 *
 * NOTES :    Contains verbose comments to illustrate how ADAS
 *            may be used from stand alone C programs.
 *
 *            This program should be linked to the adaslib libraries,
 *            compilation will resemble:
 *              g77 -I/home/adas/include adf21.c -L/home/adas/lib \
 *                     -ladaslib -o adf21.x
 *            (Note that g77 is used for compilation even though this
 *             program is written in C)
 *
 * AUTHOR:    Allan Whiteford
 *            University of Strathclyde
 *
 * DATE:      29/02/08
 *
 */

/*****************************************************************************
 xxdata_21 routine is in adaslib library so include appropriate header
*****************************************************************************/
#include "adaslib.h"

int main(void)
{
/*****************************************************************************
        Declare appopriate variables
*****************************************************************************/
        int    iunit=10 , mxbe=30     , mxtd=30     , mxtt=30;
        int    itz      , nbe         , ntdens      , nttemp;
        double svref    , beref       , tdref       , ttref;
        double be[mxbe] , tdens[mxtd] , ttemp[mxtt] , svt[mxtt];
        double sved[mxtd][mxbe];
        long lexist;
        char dsnin[81];
        char tsym[3]="  ";

/*****************************************************************************
        Set the filename appropriately
*****************************************************************************/
        strcpy(dsnin,"/home/adas/adas/adf21/bms97#h/bms97#h_c6.dat");

/*****************************************************************************
        Open the filename on a Fortran unit
*****************************************************************************/
        xxopen_ (&iunit, dsnin, &lexist, strlen(dsnin));

/*****************************************************************************
        Read the contents of the file
*****************************************************************************/
        xxdata_21_(     &iunit  , &mxbe   , &mxtd   , &mxtt   ,
                        &itz    ,  tsym   , &beref  , &tdref  ,
                        &ttref  , &svref  , &nbe    ,  be     ,
                        &ntdens ,  tdens  , &nttemp ,  ttemp  ,
                         svt    , *sved   ,  dsnin  ,  2 , 80 );

/*****************************************************************************
        Print off some example data from the file
*****************************************************************************/
        printf("%s %d %e %f %f\n",tsym,nttemp,svref,ttemp[0],ttemp[19]);

        return 0;
}
```

10.7 C example: Calculating a beam stopping coefficient

```
/*
 *      ********************** C Program: bms **********************
 *
 * Purpose:  To calculate beam effective stopping coefficients using
 *           ADAS. Originally written as an example program for
 *           Masaki Osakabe at LHD/NIFS.
 *
 *           The program calculates beam stopping coefficients at three
 *           temperatures (1keV, 2keV and 6keV) for a fixed beam energy
 *           of 165 keV / amu and a fixed plasma density of 2e13 cm-3.
 *
 *           The plasma is assumed to be 90% hydrogen and 10% carbon
 *           for all three temperatures but this can be easily changed.
 *
 * ROUTINES:
 *         ROUTINE    SOURCE    BRIEF DESCRIPTION
 *         ------------------------------------------------------------
 *         cxbms      ADAS      Calculates BMS coefficients
 *
 * NOTES :    Contains verbose comments to illustrate how ADAS
 *            may be used from stand alone C programs.
 *
 *            This program should be linked to the adas3xx and
 *            adaslib libraries, compilation will resemble:
 *              g77 -I/home/adas/include bms.c -L/home/adas/lib \
 *                     -ladas3xx -ladaslib -o adf21.x
 *            (Note that g77 is used for compilation even though this
 *             program is written in C)
 *
 * AUTHOR:    Allan Whiteford
 *            University of Strathclyde
 *
 * DATE:      29/02/08
 *
 */

/*****************************************************************************
 xxdata_21 routine is in adas3xx library so include appropriate header
*****************************************************************************/
#include "adas3xx.h"

/*****************************************************************************
 Additional required header files
*****************************************************************************/
#include <stdio.h>
#include <string.h>


int main(void)
{
/*****************************************************************************
        Declare appopriate variables
*****************************************************************************/
        int    mximp=5        , mxreqs=40      , iounit=10;
        int    nreq           , nsityp;
        double ubmeng[mxreqs] , utdens[mxreqs] , uttemp[mxreqs];
        char   dsnin[mximp][132];
        double sifrac[mximp][mxreqs];
        double bstot[mxreqs];

        int i;                  /* Variable for local use */

/*****************************************************************************
        Clear memory used for filename variable
*****************************************************************************/
        for (i=0;i<=nsityp*132;i++) (*dsnin)[i]='\0';

/*****************************************************************************
        Specify that we want 2 plasma species and 3 BMS coefficients
        calculated
*****************************************************************************/
        nsityp=2;
        nreq=3;

/*****************************************************************************
```

```
        Set filenames below for ADAS adf21 files, one for a H-beam
        into the hydrogen plasma species on to a carbon impurity
*****************************************************************************/
        strcpy(dsnin[0],"/home/adas/adas/adf21/bms97#h/bms97#h_h1.dat");
        strcpy(dsnin[1],"/home/adas/adas/adf21/bms97#h/bms97#h_c6.dat");

/*****************************************************************************
        Keep beam energy constant at 165 keV / amu for each of the
        calculated BMS coefficients
*****************************************************************************/
        ubmeng[0]=165e3;
        ubmeng[1]=165e3;
        ubmeng[2]=165e3;

/*****************************************************************************
        Keep plasma density constant at 2e13 cm-3 for each of the
        calculated BMS coefficients
*****************************************************************************/
        utdens[0]=2e13;
        utdens[1]=2e13;
        utdens[2]=2e13;

/*****************************************************************************
        Vary temperature from 1keV to 6keV for each of
        the values we get back
*****************************************************************************/
        uttemp[0]=1000e0;       /* Vary temperature from     */
        uttemp[1]=2000e0;       /* 1keV to 6keV for each of  */
        uttemp[2]=6000e0;       /* the values we get back    */

/*****************************************************************************
        90% hydrogen for each of the values we get back
*****************************************************************************/
        sifrac[0][0]=0.9;
        sifrac[0][1]=0.9;
        sifrac[0][2]=0.9;

/*****************************************************************************
        10% carbon for each of the values we get back
*****************************************************************************/
        sifrac[1][0]=0.1;
        sifrac[1][1]=0.1;
        sifrac[1][2]=0.1;

/*****************************************************************************
        Call ADAS routine to determine beam stopping
*****************************************************************************/
        cxbms_(  *dsnin,  &nsityp, &iounit, *sifrac, ubmeng, utdens,
                  uttemp, &nreq,   &mxreqs,  bstot,  132 );

/*****************************************************************************
        Print out temperature and beam stopping coefficient
*****************************************************************************/
        for (i=0;i<=2;i++)
        {
                printf("%f %e\n",uttemp[i],bstot[i]);
        }

        return 0;
}
```

10.8 C++ example: Calculating a beam stopping coefficient

```
/*
 *      ********************* C++ Program: bms ********************
 *
 * Purpose:  To calculate effective beam stopping coefficients
 *           using ADAS. Originally written as an example program
 *           for Masaki Osakabe at LHD/NIFS.
 *
 *           The program calculates beam stopping coefficients at three
 *           temperatures (1keV, 2keV and 6keV) for a fixed beam energy
 *           of 165 keV / amu and a fixed plasma density of 2e13 cm-3.
 *
 *           The plasma is assumed to be 90% hydrogen and 10% carbon
 *           for all three temperatures but this can be easily changed.
 *
 * REQUIRED CLASSES:
 *         CLASS      SOURCE    BRIEF DESCRIPTION
 *         ------------------------------------------------------
 *         cxbms      ADAS      Calculates BMS coefficients
 *
 * NOTES :    Contains verbose comments to illustrate how ADAS
 *            may be used from stand alone C++ programs.
 *
 *            This program should be linked to the adas3xx and
 *            adaslib libraries, compilation will resemble:
 *              g++ -I/home/adas/include bms.cpp -L/home/adas/lib \
 *                     -ladascpp -ladas3xx -ladaslib -lg2c -o adf21.x
 *            (Note that it is necessary to link in the 'g2c' library
 *             but this may need to be gfortran depending on local settings)
 *
 * AUTHOR:    Allan Whiteford
 *            University of Strathclyde
 *
 * DATE:      29/02/08
 *
 */

/**************************************************************************
 Include the C++ header file which defines the cxbms class.
**************************************************************************/
#include "cxbms.hpp"

int main(void)
{
/**************************************************************************
        Declare 'mybms' as an adas::cxbms object.
**************************************************************************/
        adas::cxbms mybms;

/**************************************************************************
        Specify that we want 2 plasma species and 3 BMS coefficients
        calculated
**************************************************************************/
        mybms.set_nreq(3);
        mybms.set_nsityp(2);

/**************************************************************************
        Set filenames below for ADAS adf21 files, one for a H-beam
        into the hydrogen plasma species on to a carbon impurity
**************************************************************************/
        mybms.set_file(0,"/home/adas/adas/adf21/bms97#h/bms97#h_h1.dat");
        mybms.set_file(1,"/home/adas/adas/adf21/bms97#h/bms97#h_c6.dat");

/**************************************************************************
        Keep beam energy constant at 165 keV / amu for each of the
        calculated BMS coefficients
**************************************************************************/
        mybms.set_bmeng(0,165e3);
        mybms.set_bmeng(1,165e3);
        mybms.set_bmeng(2,165e3);




/**************************************************************************
```

```
        Keep plasma density constant at 2e13 cm-3 for each of the
        calculated BMS coefficients
*****************************************************************************/
        mybms.set_dens(0,2e13);
        mybms.set_dens(1,2e13);
        mybms.set_dens(2,2e13);

/*****************************************************************************
        Vary temperature from 1keV to 6keV for each of
        the values we get back
*****************************************************************************/
        mybms.set_temp(0,1000);
        mybms.set_temp(1,2000);
        mybms.set_temp(2,6000);

/*****************************************************************************
        90% hydrogen for each of the values we get back
*****************************************************************************/
        mybms.set_frac(0,0,0.9);
        mybms.set_frac(0,1,0.9);
        mybms.set_frac(0,2,0.9);

/*****************************************************************************
        10% carbon for each of the values we get back
*****************************************************************************/
        mybms.set_frac(1,0,0.1);
        mybms.set_frac(1,1,0.1);
        mybms.set_frac(1,2,0.1);

/*****************************************************************************
        Call ADAS routine to determine beam stopping
*****************************************************************************/
        mybms.calculate();

/*****************************************************************************
        Print out temperature and beam stopping coefficient
*****************************************************************************/
        for (int i=0;i<=2;i++)
        {
                printf("%f %e\n",mybms.get_temp(i),mybms.get_bstot(i));
        }

        return 0;
}
```

Index