ADAS Subroutine xxdata_04

```
       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-----------------------------------------------------------------------
```

```
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)
```