ADAS Subroutine xxdata_23

```
      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)
```