ADAS Subroutine xxdata_00

```
       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)
```