ADAS Subroutine xxdata_12

```
      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)
```