ADAS Subroutine xxdata_15

```
       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)
```