ADAS Subroutine xxwcmt_15

```
      subroutine xxwcmt_15( iunit      ,
     &                      ndstore    , ndcmt    , ndfld   , ndsyn   ,
     &                      ndion      , ndopt    , ndlev   ,
     &                      lroot      , lsuper   ,
     &                      nfld       , isyn     , fldk    ,
     &                      lfld       ,
     &                      iz0        , iz1      , popcode ,
     &                      dsn04      , dsn18    ,
     &                      nion       , cion     , lion    ,
     &                      nopt       , copt     , lopt    ,
     &                      nlev       , config   , cterm   , ener    ,
     &                      esym       , dsnpt    ,
     &                      iptnl      , is       , tabul   , units   ,
     &                      nbsel      , nspb     , nspp    ,
     c                      ntrans     , ctrans   , wtrans  ,
     &                      ctype      , ispbr    , isppr   , iszr    ,
     &                      itg        , ipr      , iwr     ,
     &                      code       , producer , date
     &                    )
c-----------------------------------------------------------------------
c
c ****************** fortran 77 subroutine: xxwcmt_15 ******************
c
c  purpose: To write the comment section of an adf15 file .
c
c  notes:  (1) The  routine checks for the presence of field keys of
c              their synonyms, returns a value for the field if
c              present and the location & range of comment lines
c              associated with the field key if appropriate.  This
c              follows the general pattern of adf comment reading
c              subroutines.
c          (2) Analysis is carried out on the associated comment lines
c              of specific field keys to isolate and return relevant
c              information.  This is specific to the adf number.  These
c              returned data should match the data which needs to be
c              provided to enable writing of identical comment lines
c              by 'xxwcmt_15.for'
c
c  subroutine:
c
c  input : (i*4)  iunit      = unit number for input adf15 file
c  input : (i*4)  ndcnct     = maximum number of elements in connection
c  input : (i*4)  ndcmt      = maximum number of comment text lines
c  input : (i*4)  ndfld      = maximum number of search field
c  input : (i*4)  ndion      = maximum number of selective ionis coefft.
c                              inclusions
c  input : (i*4)  ndopt      = maximum number of options keys set in
c                              population code
c  input : (i*4)  ndlev      = maximum number of levels included in
c                              population calculation

c  input : (l*4)  lroot      = .true.  => output root comments
```

```
c                              .false. => do not output root comments
c  input : (l*4)  lsuper     = .true.  => output superstage comments
c                              .false. => do not output superstage comments
c  input : (i*4)  nfld       = number of fields for adf15 comments
c  input : (i*4)  isyn()     = number of synonyms for fields
c                              1st dim: field index (1->nfld)
c  input :  (c*40) fldk(,)    = field keys
c                              1st dim: field index (1->nfld)
c                              2nd dim: synonymn index (1->isyn())
c  input : (l*4)  lfld()     = .true.  => field  available for comments
c                            = .false. => field not available
c                              1st dim: field index (1->nfld)
c  input : (i*4)  iz0        = nuclear charge
c  input : (i*4)  iz1        = emitting ion charge+1
c  input : (c*7)  popcode    = propulation processing code
c  input : (c*120)dsn04      = adf04 file used by population code
c  input : (c*120)dsn18      = adf18 map file used to access projection
c  input : (i*4)  nion       = number of selective ionis. coefft.
c                             inclusions in population calculation
c  input : (c*5)  cion()     = selec. ionis coefft. spec as (ispb,ispp)
c                              1st dim: ionis coefft. list index
c  input : (l*1)  lion()     = .true. => included
c                            = .false.=> not included
c                              1st dim: ionis coefft. list index
c  input : (i*4)  nopt       = number of option keys present for
c                              population calculation
c  input : (c*6)  copt()     = option specification strings as l*****
c                              1st dim: option list index
c  input : (l*1)  lopt()     = .true. => set
c                            = .false.=> not set
c                              1st dim: option list index
c  input : (i*4)  nlev       = number of energy levels included in
c                              population calculation
c  input : (c*19) config()   = configuration string
c                              1st dim: level list index
c  input : (c*14) cterm()    = term/level specification string
c                              1st dim: level list index
c  input : (r*8)  ener()     = energy level relative to lowest (cm^-1)
c                              1st dim: level list index
c  input : (c*2)  esym       = element symbol
c  input : (c*120)dsnpt      = parent file template used to create
c                              current child partition (blank if root)
c  input : (i*4)  iptnl      = current partition level
c  input : (i*4)  is         = superstage label
c  input : (c*40) tabul      = adf15 quantity tabulated specification
c  input : (c*40) units      = adf15 units use specification
c  input : (i*4)  nspb       = number of excitation (base) drivers for
c                              superstage (= icnctv(is))
c  input : (i*4)  nspp       = number of recombination (parent) drivers
c                              for superstage (=icnctv(is+1))
c  input : (i*4)  nbsel      = number of emissivity line blocks in the
c                              adf15 file
c  input : (i*4)  ntrans     = number of distinct transitions in the
```

```
c                              adf15 file
c  input : (c*29) ctrans()   = transition specification string
c                              1st dim: emissivity index (1->ntrans)
c  input : (r*8)  wtrans()   = transition wavelength
c
c  input : (c*5)  ctype()    = transition type for each line block
c                              1st dim: index of block in adf15 file
c  input : (i*4)  ispbr()    = base driver index for each line block
c                              1st dim: index of block in adf15 file
c  input : (i*4)  isppr()    = parent driver index for each line block
c                              1st dim: index of block in adf15 file
c  input : (i*4)  iszr()     = ion charge relating to each line
c                              1st dim: index of block in adf15 file
c  input : (i*4)  itg()      = transition group attribution of
c                              emissivity  line block
c                              1st dim: index of block in adf15 file
c  input : (i*4)  ipr()      = power ranking of emissivity line
c                              block (note power is a composite
c                              attribute of a transition group)
c                              1st dim: index of block in adf15 file
c  input : (c*7)  code       = ADAS code which generated the
c                              superstage adf15 file
c  input : (c*30) producer   = producer of the adf15 file
c  input : (c*8)  date       = date of creattion of the superstage
c                              adf15 file
c
c routines:
c         routine    source    brief description
c         ----------------------------------------------------------
c         i4unit     adas      fetch unit number for output of messages
c         r8fctn     adas      convert string to real number
c         xxslen     adas      find string less front and tail blanks
c         xxcase     adas      convert a string to upper or lower case
c         xxordr     adas      order a real vector retaining indexing
c
c
c author:  H. P. Summers, University of Strathclyde
c          ja7.08
c          tel. 0141-548-4196
c
c date:    19/05/06
c
c
c version  : 1.1
c date     : 19-05-2006
c modified : H P Summers
c              - first version.
c
c-----------------------------------------------------------------------
      CHARACTER*5         CION(NDION)
      CHARACTER*7         CODE
      CHARACTER*19        CONFIG(NDLEV)
      CHARACTER*6         COPT(NDOPT)
```

```
CHARACTER*14        CTERM(NDLEV)
CHARACTER*29        CTRANS(NDSTORE)
CHARACTER*5         CTYPE(NDSTORE)
CHARACTER*8         DATE
CHARACTER*80        DSN04,       DSN18
CHARACTER*120       DSNPT
CHARACTER*2         ESYM
CHARACTER*40        FLDK(NDFLD,NDSYN)
CHARACTER*7         POPCODE
CHARACTER*30        PRODUCER
CHARACTER*60        TABUL,       UNITS
INTEGER             IPR(NDSTORE),             IPTNL,       IS
INTEGER             ISPBR(NDSTORE),           ISPPR(NDSTORE)
INTEGER             ISYN(NDFLD), ISZR(NDSTORE)
INTEGER             ITG(NDSTORE),             IUNIT
INTEGER             IWR(NDSTORE),             IZ0,         IZ1
INTEGER             NBSEL,       NDCMT,       NDFLD,       NDION
INTEGER             NDLEV,       NDOPT,       NDSTORE,     NDSYN
INTEGER             NFLD,        NION,        NLEV,        NOPT
INTEGER             NSPB,        NSPP,        NTRANS
LOGICAL             LFLD(NDFLD), LION(NDION), LOPT(NDOPT), LROOT
LOGICAL             LSUPER
REAL*8              ENER(NDLEV), WTRANS(NDSTORE)
```