

## ADAS Subroutine xxwcmt\_40

```
subroutine xxwcmt_40( 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    , dsnt   ,
&                    iptnl   , is     , tabul  , units  ,
&                    nbsel   , nspb  , nspp   ,
&                    nrange  , npixr  , wvminr , wvmaxr ,
&                    ilzr    , ihzr  ,
&                    ctype   , ispbr  , isppr  ,
&                    irg     , ipr    , iwr    ,
&                    code    , producer , date
&                    )
```

```
-----
C
C ***** fortran 77 subroutine: xxwcmt_40 *****
C
C purpose: To write the comment section of an adf40 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
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 adf40 file
C input : (i*4) ndstore   = maximum number of input data-blocks
C                          that can be stored
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
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 adf40 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   = population 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     = adf40 quantity tabulated specification
c input : (c*40) units     = adf40 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)  nbssel    = number of emissivity line blocks in the

```

```

c                               adf40 file
c input : (i*4)  nrange         = number of distinct wave length ranges
c                               in the  adf40 file
c
c input : (i*4)  npixr()        = number of pivels in wavelength range
c                               1st dim: wavelength range count
c input : (r*8)  wvminr()       = min. wavelength (A) of wavelength range
c                               1st dim: wavelength range count
c input : (r*8)  wvmaxr()       = max. wavelength (A) of wavelength range
c                               1st dim: wavelength range count
c input : (i*4)  ilzr()         = lowest charge state contributing to
c                               the wavelength range
c                               1st dim: wavelength range count
c input : (i*4)  ihzr()         = highest charge state contributing to
c                               the wavelength range
c                               1st dim: wavelength range count
c input : (c*5)  ctype()        = transition type for each line block
c                               1st dim: index of block in adf40 file
c input : (i*4)  ispbr()        = base driver index for each line block
c                               1st dim: index of block in adf40 file
c input : (i*4)  isppr()        = parent driver index for each line block
c                               1st dim: index of block in adf40 file
c input : (i*4)  iszr()         = ion charge relating to each line
c                               1st dim: index of block in adf40 file
c input : (i*4)  irg()          = transition group attribution of
c                               emissivity line block
c                               1st dim: index of block in adf40 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 adf40 file
c input : (c*7)  code           = ADAS code which generated the
c                               superstage adf40 file
c input : (c*30) producer       = producer of the adf40 file
c input : (c*8)  date           = date of creattion of the superstage
c                               adf40 file

```

c routines:

routine	source	brief description
i4unit	adas	fetch unit number for output of messages
r8fctn	adas	convert string to real number
xxslen	adas	find string less front and tail blanks
xxcase	adas	convert a string to upper or lower case
xxordr	adas	order a real vector retaining indexing

```

c author: h. p. summers, university of strathclyde
c         ja7.08
c         tel. 0141-548-4196

```

```

c date: 15/06/06

```

c  
c version : 1.1  
c date : 15-06-2006  
c modified : H P Summers  
c - first version.

c

c

-----

CHARACTER*5	CI	ON (NDION)	
CHARACTER*7	CODE		
CHARACTER*19	CONFIG (NDLEV)		
CHARACTER*6	COPT (NDOPT)		
CHARACTER*14	CTERM (NDLEV)		
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	IHZR (NDSTORE),	ILZR (NDSTORE)	
INTEGER	IPR (NDSTORE),	IPTNL	
INTEGER	IRG (NDSTORE),	IS	
INTEGER	ISPBR (NDSTORE),	ISPPR (NDSTORE)	
INTEGER	ISYN (NDFLD), IUNIT,	IWR (NDSTORE)	
INTEGER	IZ0,	IZ1,	NBSEL, NDCMT
INTEGER	NDFLD,	NDION,	NDLEV, NDOPT
INTEGER	NDSTORE,	NDSYN,	NFLD, NION
INTEGER	NLEV,	NOPT,	NPIXR (NDSTORE)
INTEGER	NRANGE,	NSPB,	NSPP
LOGICAL	LFLD (NDFLD), LION (NDION),	LOPT (NDOPT),	LROOT
LOGICAL	LSUPER		
REAL*8	ENER (NDLEV),	WVMAXR (NDSTORE)	
REAL*8	WVMINR (NDSTORE)		