ADAS Subroutine xxdata_25

```
      subroutine xxdata_25( iunit  , a25fmt , dsname ,
     &                      ndtem  , ndden  , ndrep  , ndcor , nddiel ,
     &                      nddef  , ndimp  , ndein  , ndzef ,
     &                      iz0    , iz1    , outfmt ,
     &                      exfile , cxfile ,
     &                      ndens  , id_ref , densa  , denpa , denimpa,
     &                      deniona,
     &                      ntemp  , it_ref , tea    , tpa   , timpa  ,
     &                      tiona  ,
     &                      nzef   , iz_ref , zefa   ,
     &                      nbeam  , ib_ref , bmena  , denha , bmfra  ,
     &                      nimp   , im_ref , zimpa  , amimpa, frimpa ,
     &                      ts     , w      , w1     ,
     &                      cion   , cpy    , nip    , intd  , iprs   ,
     &                      ilow   , ionip  , nionip , ilprs , ivdisp ,
     &                      nmin   , nmax   , imax   , nrep  , wbrep  ,
     &                      jdef   , def    ,
     &                      jcor   , cor    , jmax   , epsil , fij    ,
     &                      wij
     &                    )
c-----------------------------------------------------------------------
c
c  ***************** fortran77 subroutine: xxdata_25 *******************
c
c  purpose:  To fetch  data  from  an adf25 driver dataset.
c
c  Notes:
c
c  Subroutine:
c
c   input : (i*4)  iunit    = unit number for input adf01 file.
c   input : (c*8)  a25fmt   = subdirectory type of adf25 to be read.
c   input : (c*80) dsname   = file name of adf25 format to be read.
c
c   input : (i*4)  ndtem    = maximum number of electron temperatures
c   input : (i*4)  ndden    = maximum number of electron densities
c   input : (i*4)  ndrep    = maximum number of representative n-shells
c   input : (i*4)  ndcor    = maximum number of DR bethe corrections
c   input : (i*4)  nddiel   = maximum number of DR core transitions
c   input : (i*4)  nddef    = maximum number of quantum defects
c   input : (i*4)  ndimp    = maximum number of plasma impurities
c   input : (i*4)  ndein    = maximum number of beam energies
c   input : (i*4)  ndzef    = maximum number of z effectives
c
c   input : (i*4)  iz0      = nuclear charge of bundle-n ion
c   input : (i*4)  iz1      = recombining ion charge of bundle-n ion
c   input : (c*5)  outfmt   = format of output ADAS data format for final
c                             results
c   input : (c*80) cxfile   = file name for charge exchange data input
c   input : (c*80) exfile   = file name for map of proj. matrix output
c
c   input : (i*4)  ndens    = number of electron densities
```

```
c   input : (i*4)  id_ref   = reference electron density pointer in vectors
c   input : (i*4)  densa()  = plasma electron density vector (cm-3)
c                             1st dim: index of electron density
c   input : (i*4)  denpa()  = plasma H+ density vector (cm-3)
c                             1st dim: index of electron density
c   input : (i*4)  denimpa()= plasma mean impurity ion density (cm-3)
c                             1st dim: index of electron density
c   input : (i*4)  deniona()= total ion density (plasma+impurity) (cm-3)
c                             1st dim: index of electron density
c
c   input : (i*4)  ntemp    = number of electron temperatures
c   input : (i*4)  id_ref   = reference electron temp. pointer in vectors
c   input : (i*4)  tea()    = plasma electron temp. vector (K)
c                             1st dim: index of electron temperature
c   input : (i*4)  tpa()    = plasma H+ temp. vector (K)
c                             1st dim: index of electron temperature
c   input : (i*4)  timpa()  = plasma mean impurity ion temp (K)
c                             1st dim: index of electron temperature
c   input : (i*4)  tiona()  = mean ion temp (plasma+impurity) (K)
c                             1st dim: index of electron temperature
c
c   input : (i*4)  nzef     = number of plasma zeff
c   input : (i*4)  iz_ref   = reference zeff pointer in vector
c   input : (i*4)  zefa()   = plasma zeff vector
c                             1st dim: index of zeff
c
c   input : (i*4)  nbeam    = number of beam energies
c   input : (i*4)  ib_ref   = reference beam energy pointer in vectors
c   input : (i*4)  bmena()  = beam energy vector (ev/amu)
c                             1st dim: index of beam energies
c   input : (i*4)  denha()  = beam H+ density vector (cm-3)
c                             1st dim: index of beam energies
c   input : (i*4)  bmfra()  = fractions of beam at each energy
c                             1st dim: index of beam energies
c
c   input : (i*4)  nimp     = number of plasma impurities (excl.h+)
c   input : (i*4)  im_ref   = reference impurity pointer in vectors
c   input : (r*8)  zimpa()  = impurity species charge
c                             1st dim: index of impurity
c   input : (r*8)  amimpa() = atomic mass number of impurity species
c                             1st dim: index of impurity
c   input : (r*8)  frimpa() = fraction of impurity (normalised to 1)
c                             1st dim: index of impurity
c
c   input : (r*8)  ts       = external radiation field temperature (K)
c   input : (r*8)  w        = general radiation dilution factor
c   input : (i*4)  w1       = external radiation field dilution factor
c                             for photo-ionisation form the ground level.
c
c   input : (r*8)  cion     = adjustment multiplier for ground ionis.
c   input : (r*8)  cpy      = adjustment multiplier for VR xsects.
c   input : (i*4)  nip      = range of delta n for IP xsects. (le.4)
c   input : (i*4)  intd     = order of Maxw. quad. for IP xsects.(le.3)
```

```
c   input : (i*4)  iprs      = 0  => default to VR xsects. beyond nip range
c                              1  => use PR xsects. beyond nip range
c   input : (i*4)  ilow      = 0  => no special low level data accessed
c                              1  => special low level data accessed
c   input : (i*4)  ionip     = 0 =>   no ion impact collisions included
c                              1 =>ion impact excit. and ionis. included
c   input : (i*4)  nionip    = range of delta n for ion impact
c                              excitation xsects.
c   input : (i*4)  ilprs     = 0 => default to vainshtein xsects.
c                              1 => use lodge-percival-richards xsects.
c   input : (i*4)  ivdisp    = 0 => ion impact at thermal Maxw. energies
c                              1 => ion impact at displaced thermal
c                                   energies according to the neutral
c                                   beam energy parameter
c                                * if(ivdisp=0 then special low level
c                                  data for ion impact is not substituted -
c                                  only vainshtein and lodge et al.
c                                  options are open.  Electron impact
c                                  data substitution does occur.
c   input : (i*4)  nmin      = lowest n-shell for population structure
c   input : (i*4)  nmax      = highest n-shell for population structure
c   input : (i*4)  imax      = number of representative n-shells
c   input : (i*4)  nrep()    = representative n-shells
c                              1st dim: index of representative n-shell
c   input : (r*8)  wbrep()   = dilution factors for nmin->nrep() trans.
c                              1st dim: index of representative n-shell
c   input : (i*4)  jdef      = number of n-shell quantum defects
c   input : (r*8)  def()     = quantum defects for n-shells
c                              1st dim: index of n-shell quantum defects
c                                       upwards from nmin
c   input : (i*4)  jcor      = number of DR Bethe correction factors
c   input : (r*8)  cor()     = DR Bethe correction factors
c                              1st dim: index of correction factor
c   input : (i*4)  jmax      = number of DR core transitions
c   input : (r*8)  epsil()   = reduced energy of core transition
c                              [delta Eij/I_H=(z+1)^2*epsil()]
c                              1st dim: index of DR core transition
c   input : (r*8)  fij()     = absorption oscillator strength for
c                              DR core transition
c                              1st dim: index of DR core transition
c   input : (r*8)  wij()     = dilution factor for DR core transition
c                              1st dim: index of DR core transition
c
c  Routines:
c
c     routine    source     brief description
c     -------------------------------------------------------------
c     xxcase     adas       convert string to upper or lower case
c     xxslen     adas       locate first and last char. of string
c
c
c  Author   : Hugh Summers
c  Date     : 23-05-2007
```

```
c
c
c  Version  : 1.1
c  Date     : 23-05-2007
c  Author   : Hugh Summers
c               - First release.
c
c  Version  : 1.2
c  Date     : 21-11-2007
c  Author   : Martin O'Mullane
c               - Increase number of lines to 150 to accommodate full
c                 range of energies/temperatures/densities of
c                 the dataformat.
c
c-----------------------------------------------------------------------
      CHARACTER*8         A25FMT
      CHARACTER*(*)       CXFILE,      DSNAME,      EXFILE
      CHARACTER*5         OUTFMT
      INTEGER             IB_REF,      ID_REF,      ILOW,        ILPRS
      INTEGER             IMAX,        IM_REF,      INTD,        IONIP
      INTEGER             IPRS,        IT_REF,      IUNIT,       IVDISP
      INTEGER             IZ0,         IZ1,         IZ_REF,      JCOR
      INTEGER             JDEF,        JMAX,        NBEAM,       NDCOR
      INTEGER             NDDEF,       NDDEN,       NDDIEL,      NDEIN
      INTEGER             NDENS,       NDIMP,       NDREP,       NDTEM
      INTEGER             NDZEF,       NIMP,        NIONIP,      NIP
      INTEGER             NMAX,        NMIN,        NREP(NDREP+1)
      INTEGER             NTEMP,       NZEF
      REAL*8              AMIMPA(NDIMP),            BMENA(NDEIN)
      REAL*8              BMFRA(NDEIN),             CION
      REAL*8              COR(NDCOR),  CPY,         DEF(NDDEF)
      REAL*8              DENHA(NDEIN),             DENIMPA(NDDEN)
      REAL*8              DENIONA(NDDEN),           DENPA(NDDEN)
      REAL*8              DENSA(NDDEN),             EPSIL(NDDIEL)
      REAL*8              FIJ(NDDIEL), FRIMPA(NDIMP)
      REAL*8              TEA(NDTEM),  TIMPA(NDTEM)
      REAL*8              TIONA(NDTEM),             TPA(NDTEM),  TS
      REAL*8              W,           W1,          WBREP(NDREP)
      REAL*8              WIJ(NDDIEL), ZEFA(NDZEF), ZIMPA(NDIMP)
```