

ADAS Subroutine xxdata_11

```
subroutine  xxdata_11( iunit  , iclass ,
&                    isdimd ,  iddimd , itdimd ,
&                    ndptnl , ndptn  , ndptnc , ndcnct ,
&                    iz0    , islmin , islmax ,
&                    nptnl  , nptn   , nptnc  ,
&                    iptnla , iptna  , iptnca ,
&                    ncnct  , icnctv ,
&                    iblmx  , ismax  , dnr_ele, dnr_ams,
&                    isppr  , ispbr  , isstgr ,
&                    idmax  , itmax  ,
&                    ddens  , dtev   , drcof  ,
&                    lres   , lstan  , lptn
&                    )
```

```
C-----
C
C ***** fortran77 subroutine: xxdata_11 *****
C
C purpose: to read a complete adf11 file, check its class and
C          determine its standard, resolved and partition organisation.
C
C calling program: various
C
C notes:   (1) A 'standard' adf11 file contains gcr data between one
C            whole ionisation stage and another whole ionisation
C            stage.
C            A 'resolved' (or partial) adf11 file contains gcr data
C            between a set of metastables of one ionisation stage
C            and a set of metastables of another ionisation stage.
C            A resolved file is distinguished from a standard file
C            by the presence of a 'connection vector' in the adf11
C            data file header lines.
C            The connection vector specifies the number of meta-
C            stables in each ionisation stage which are coupled
C            together by gcr data.
C            (2) A 'partitioned' adf11 file contains gcr data between
C            clumps of ionisation stages or metastables or comb-
C            inations of the two called 'partitions'.
C            A 'partition level' is a specification of the
C            partitions which span all the ionisation stages (and
C            metastables) of an element. Successive partition
C            levels give a heirarchy corresponding to larger
C            partitions and greater clumping.
C            A 'superstage' is a set of partitions which are close-
C            coupled.
C            There are thus equivalences :
C                ionisation stage - superstage
C                metastable       - partition
C                ion charge       - superstage index
C            A partitioned adf11 file may be standard (with each
C            superstage comprising only one partition) or resolved.
C            A partitioned file is distinguished by the presence of
C            'partition specification block' in the adf11 data
```



```

c          'iprtr' and 'igrd' in old notation are now substituted
c          by 'isprr' and 'ispbr' respectively internally and in
c          external naming.
c
c
c          subroutine:
c
c          input : (i*4)  iunit      = unit to which input file is allocated
c          input : (i*4)  iclass     = class of data (1 - 12 ):
c                               1-acd, 2-scd, 3-ccd, 4-prb, 5-prc
c                               6-qcd, 7-xcd, 8-plt, 9-pls,10-zcd
c                               11-ycd,12-ecd
c
c          input : (i*4)  isdimd    = maximum number of (sstage, parent, base)
c                               blocks in isonuclear master files
c          input : (i*4)  iddimd    = maximum number of dens values in
c                               isonuclear master files
c          input : (i*4)  itdimd    = maximum number of temp values in
c                               isonuclear master files
c          input : (i*4)  ndptnl    = maximum level of partitions
c          input : (i*4)  ndptn     = maximum no. of partitions in one level
c          input : (i*4)  ndptnc    = maximum no. of components in a partition
c          input : (i*4)  ndcnct    = maximum number of elements in connection
c                               vector
c
c          output: (i*4)  iz0       = nuclear charge
c          output: (i*4)  islmin    = minimum ion charge + 1
c                               (generalised to connection vector index)
c          output: (i*4)  islmax    = maximum ion charge + 1
c                               (note excludes the bare nucleus)
c                               (generalised to connection vector index
c                               and excludes last one which always remains
c                               the bare nucleus)
c          output: (i*4)  nptnl     = number of partition levels in block
c          output: (i*4)  nptn( )   = number of partitions in partition level
c                               1st dim: partition level
c          output: (i*4)  nptnc( , ) = number of components in partition
c                               1st dim: partition level
c                               2nd dim: member partition in partition level
c          output: (i*4)  iptnla( ) = partition level label (0=resolved root,1=
c                               unresolved root)
c                               1st dim: partition level index
c          output: (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          output: (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          output: (i*4)  ncnct     = number of elements in connection vector
c          output: (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
c output: (i*4)  iblrx      = number of (sstage, parent, base)
c                   blocks in isonuclear master file
c output: (i*4)  ismax      = number of charge states
c                   in isonuclear master file
c                   (generalises to number of elements in
c                   connection vector)
c output: (c*12) dnr_ele    = CX donor element name for iclass = 3 or 5
c                   (blank if unset)
c output: (r*8)  dnr_ams    = CX donor element mass for iclass = 3 or 5
c                   (0.0d0 if unset)
c output: (i*4)  isppr()    = 1st (parent) index for each partition block
c                   1st dim: index of (sstage, parent, base)
c                   block in isonuclear master file
c output: (i*4)  ispbr()    = 2nd (base) index for each partition block
c                   1st dim: index of (sstage, parent, base)
c                   block in isonuclear master file
c output: (i*4)  isstgr()   = s1 for each resolved data block
c                   (generalises to connection vector index)
c                   1st dim: index of (sstage, parent, base)
c                   block in isonuclear master file
c
c output: (i*4)  idmax      = number of dens values in
c                   isonuclear master files
c output: (i*4)  itmax      = number of temp values in
c                   isonuclear master files
c output: (r*8)  ddens()    = log10(electron density(cm-3)) from adf11
c output: (r*8)  dtev()     = log10(electron temperature (eV) from adf11
c output: (r*8)  drcof(,,) = if(iclass <=9):
c                   log10(coll.-rad. coefft.) from
c                   isonuclear master file
c                   if(iclass >=10):
c                   coll.-rad. coefft. from
c                   isonuclear master file
c                   1st dim: index of (sstage, parent, base)
c                   block in isonuclear master file
c                   2nd dim: electron temperature index
c                   3rd dim: electron density index
c
c output: (l*4)  lres       = .true. => partial file
c                   = .false. => not partial file
c output: (l*4)  lstan      = .true. => standard file
c                   = .false. => not standard file
c output: (l*4)  lptn      = .true. => partition block present
c                   = .false. => partition block not present
c
c routines:
c   routine      source      brief description
c   -----
c   i4unit       adas        fetch unit number for output of messages

```

```

c      i4fctn      adas      convert string to integer form
c      xfelem      adas      return element name given nuclear charge
c      xxword      adas      extract position of number in buffer
c      xxslen      adas      find string less front and tail blanks
c      xxcase      adas      convert a string to upper or lower case
c      xxrptn      adas      analyse an adf11 file partition block

```

```

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

```

```

c date:    04/10/06

```

```

c version: 1.1 date: 04/10/2006

```

```

c modified: hugh summers

```

```

c - first edition.

```

```

c version: 1.2 date: 21/01/2007

```

```

c modified: Allan Whiteford

```

```

c - Commented out warning about lack of iclass,
c           all of the present ADAS files do not contain
c           this information
c           (first commit to CVS)

```

```

c version: 1.3 date: 08/03/2007

```

```

c modified: Hugh Summers

```

```

c - adjustments for revised ecd formats.

```

```

c           charge exchange donor/donor mass checks and
c           dnr_ele, dnr_ams added to parameter return.

```

```

-----
CHARACTER*12      DNR_ELE
INTEGER           IBLMX,          ICLASS,          ICNCTV (NDCNCT)
INTEGER           IDDIMD,         IDMAX,           IPTNA (NDPTNL, NDPTN)
INTEGER           IPTNCA (NDPTNL, NDPTN, NDPTNC)
INTEGER           IPTNLA (NDPTNL) ,          IS1MAX,          IS1MIN
INTEGER           ISDIMD,         ISMAX,           ISPBR (ISDIMD)
INTEGER           ISPPR (ISDIMD) ,          ISSTGR (ISDIMD)
INTEGER           ITDIMD,         ITMAX,           IUNIT,          IZ0
INTEGER           NCNCT,          NDCNCT,         NDPTN,          NDPTNC
INTEGER           NDPTNL,         NPTN (NDPTNL)
INTEGER           NPTNC (NDPTNL, NDPTN) ,          NPTNL
LOGICAL          LPTN,           LRES,          LSTAN
REAL*8           DDENS (IDDIMD) ,          DNR_AMS
REAL*8           DRCOF (ISDIMD, ITDIMD, IDDIMD)
REAL*8           DTEV (ITDIMD)

```