ADAS Subroutine h4data

```
C
       subroutine h4data( iunit  , ixdim  , itdim   , isdim  ,
     &                    dsfull , indxref , title  , cameth ,
     &                    z0     , z       , zeff   ,
     &                    n1     , l1      , eb1    ,
     &                    n2     , l2      , eb2    ,
     &                    isp    , lp      , xjp    ,
     &                    ist1   , lt1     , xj1    , xjt1   ,
     &                    ist2   , lt2     , xj2    , xjt2   ,
     &                    neqv1  , fpc1    , neqv2  , fpc2   ,
     &                    aval   ,
     &                    xmax   , iextwf  , ijucys , isrch  ,
     &                    nshell , nlqs    , alfaa  ,
     &                    nx     , xa      ,
     &                    nt     , tea     ,
     &                    cresol
     &                  )
c-----------------------------------------------------------------------
c
c  ********************* fortran77 subroutine h4data ******************
c
c  purpose: to refresh a data index from an adas804 archive. reads
c           in the index code a-effective potential Born, b-impact
c           parameter and the the rest of the data as appropriate.
c
c  calling program: adas804
c
c  subroutine:
c
c  input : (i*4)  iunit    = unit to be used for reading file
c  input : (i*4)  ixdim    = maximum dimension for X  array
c  input : (i*4)  itdim    = maximum dimension for Te array
c  input : (i*4)  isdim    = maximum dimension for shell vectors
c
c  input : (c*80) dsfull   = the users' chosen archive file name.
c  input : (i*4)  indxref  = the index number to refresh from.
c
c  output: (c*40) title    = transition title in the archive file.
c  output: (c*1)  cameth   = the tag distinguishing the type of
c                            analysis: a - Born, b- IP
c  output: (r*8)  z0       = nuclear charge
c  output: (r*8)  z        = ion charge
c  output: (r*8)  zeff     = ion charge +1
c  output: (i*4)  n1       = lower n-shell of transition
c  output: (i*4)  l1       = lower l-shell of transition
c  output: (r*8)  eb1      = binding energy (Ryd) in lower level
c  output: (i*4)  n2       = upper n-shell of transition
c  output: (i*4)  l2       = upper l-shell of transition
c  output: (r*8)  eb2      = binding energy (Ryd) in upper level
c  output: (i*4)  isp      = 2*Sp+1 for parent
c  output: (i*4)  lp       = Lp for parent
c  output: (r*8)  xjp      = Jp for parent (if 'ic' coupling)
```

```
c  output: (i*4)  ist1     = 2*S+1 for lower state
c  output: (i*4)  lt1      = L for lower state
c  output: (r*8)  xj1      = j for lower state
c  output: (r*8)  xjt1     = J for lower state
c  output: (i*4)  ist2     = 2*S'+1 for upper state
c  output: (i*4)  lt2      = L' for upper state
c  output: (r*8)  xj2      = j' for upper state
c  output: (r*8)  xjt2     = J' for upper state
c  output: (i*4)  neqv1    = no. of equiv. electrons for lower shell.
c  output: (r*8)  fpc1     = fract. parentage for lower state
c  output: (i*4)  neqv2    = no. of equiv. electrons for upper shell.
c  output: (r*8)  fpc2     = fract. parentage for upper state
c  output: (i*4)  aval     = A-value (sec-1) if dipole; else -ve
c  output: (i*4)  xmax     = range of numerical wave functions
c  output: (i*4)  iextwf   = 0 => calculate radial wave functions
c                          = 1 => read in radial wave functions
c  output: (i*4)  ijucys   = -1 => Jucys potential form adopted
c                          = 0  => Slater potential form adopted
c  output: (i*4)  isrch    = 0 => fcf6 search for energy eigenvalue
c                          = 1 => fcf6 search for scaling parameters
c  output: (i*4)  nshell   = number of screening shells
c  output: (i*4)  nlqs     = 1000*n+100*l+iq for each screening shell
c                              1st dim: screening shell index
c  output: (i*4)  alfaa    = scaling factor for each screening shell
c                              1st dim: index for lower & upper state
c                              2nd dim: index over screening shells
c  output: (i*4)  nx       = number of incident electron energies
c  output: (i*4)  xa()     = threshold parameter values
c  output: (i*4)  nt       = number of electron temperatures
c  output: (i*4)  tea()    = electron temperatures (K)
c  output: (i*4)  cresol   = 'ic' => transition between J-levels
c                          = 'ls' => transition between terms
c
c
c  routines:
c          routine    source    brief description
c          ------------------------------------------------------------
c          i4unit     adas      fetch unit number for output of messages
c          xxword     adas      extract position of number in buffer
c          xxcase     adas      change string to all upper or lower case
c
c  author:  Hugh P. Summers, University of Strathclyde
c           JA7.08
c           Tel.: +44 (0)141-548-4196
c
c  date:    24 February 2003
c
c
c  version: 1.1   Hugh Summers  24/02/03
c  modified:      first release
c
c-----------------------------------------------------------------------
      CHARACTER           CAMETH
```

```
CHARACTER*2         CRESOL
CHARACTER*80        DSFULL
CHARACTER*40        TITLE
INTEGER             IEXTWF,       IJUCYS,       INDXREF,      ISDIM
INTEGER             ISP,          ISRCH,        IST1,         IST2
INTEGER             ITDIM,        IUNIT,        IXDIM,        L1
INTEGER             L2,           LP,           LT1,          LT2
INTEGER             N1,           N2,           NEQV1,        NEQV2
INTEGER             NLQS(ISDIM),  NSHELL,       NT,           NX
REAL*8              ALFAA(2,ISDIM),             AVAL,         EB1
REAL*8              EB2,          FPC1,         FPC2
REAL*8              TEA(ITDIM),   XA(IXDIM),    XJ1,          XJ2
REAL*8              XJP,          XJT1,         XJT2,         XMAX
REAL*8              Z,            Z0,           ZEFF
```