ADAS Subroutine xxdata_24

```
       subroutine xxdata_24( iunit  , dsname ,
     &                       nstore , nedim  ,
     &                       esym   , iz0    ,
     &                       nbsel  , isela  ,
     &                       iz     , iz1    ,
     &                       cdonor , crecvr , cfstat , ctype,
     &                       alph0  ,
     &                       iea    ,
     &                       teea   , scx
     &                     )
C-----------------------------------------------------------------------
C
C  ***************** FORTRAN77 SUBROUTINE: xxdata_24 ******************
C
C  PURPOSE:  To  fetch  data  from  input charge exchange cross-section
C            data for  given donor and receiver ions.
C
C  CALLING PROGRAM: ADAS509/SSCX
C
C  DATA:     Up to 'nstore' sets (data-blocks) of data may be  read from
C            the file - each block forming a complete set of  cross-
C            section values for given collision energies.
C            Each data-block  is  analysed independently of any  other
C            data-block.
C
C            The units used in the data file are taken as follows:
C
C            Collision energies  : eV/amu
C            Cross-section       : cm**2
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  IUNIT    = UNIT TO WHICH INPUT FILE IS ALLOCATED.
C  INPUT : (C*44) DSNAME   = FILENAME OF DATA SET BEING READ
C
C  INPUT : (I*4)  NSTORE   = MAXIMUM NUMBER  OF  INPUT DATA-BLOCKS  THAT
C                            CAN BE STORED.
C  INPUT : (I*4)  NEDIM    = MAX NUMBER OF COLLISION ENERGIES ALLOWED
C
C  OUTPUT: (C*2)  ESYM     = READ - RECEIVING ION - ELEMENT SYMBOL
C  OUTPUT: (I*4)  IZ0      = READ - RECEIVING ION - NUCLEAR CHARGE
C
C  OUTPUT: (I*4)  NBSEL    = NUMBER OF DATA-BLOCKS ACCEPTED & READ IN.
C  OUTPUT: (I*4)  ISELA()  = READ - DATA-SET DATA-BLOCK ENTRY INDICES
C                            DIMENSION: DATA-BLOCK INDEX
C
C  OUTPUT: (I*4)  IZ       = READ - RECEIVED ION -  CHARGE
C  OUTPUT: (I*4)  IZ1      = READ - RECEIVING ION - CHARGE
C
C  OUTPUT: (C*9)  CDONOR() = READ - DONOR ION IDENTIFICATION
C                            DIMENSION: DATA-BLOCK INDEX
C  OUTPUT: (C*9)  CFCODE() = READ - RECEIVER ION IDENTIFICATION
```

```
C                               DIMENSION: DATA-BLOCK INDEX
C  OUTPUT: (C*10) CFSTAT() = READ - FINAL STATE SPECIFICATION
C                               DIMENSION: DATA-BLOCK INDEX
C  OUTPUT: (C*2)  CTYPE()  = READ - CROSS-SECTION TYPE
C                               DIMENSION: DATA-BLOCK INDEX
C
C  OUTPUT: (R*8)  ALPH0()  = READ - LOW ENERGY EXTRAPOLATION PARM.
C                               DIMENSION: DATA-BLOCK INDEX
C
C
C  OUTPUT: (I*4)  IEA()    = READ - NUMBER OF COLLISION ENERGIES
C                               DIMENSION: DATA-BLOCK INDEX
C
C  OUTPUT: (R*8)  TEEA(,)  = READ - COLLISION ENERGIES (UNITS: eV/AMU)
C                               1st DIMENSION: COLLISION ENERGY INDEX
C                               2nd DIMENSION: DATA-BLOCK INDEX
C
C  OUTPUT: (R*8)  SCX(,)   =READ - FULL SET OF COLLISION CROSS-
C                                       SECTION VALUES (cm**2)
C                               1st DIMENSION: COLLISION ENERGY INDEX
C                               2nd DIMENSION: DATA-BLOCK INDEX
C
C          (I*4)  I4EIZ0   = FUNCTION - (SEE ROUTINES SECTION BELOW)
C          (I*4)  I4FCTN   = FUNCTION - (SEE ROUTINES SECTION BELOW)
C          (I*4)  I4UNIT   = FUNCTION - (SEE ROUTINE SECTION BELOW)
C          (I*4)  IBLK     = ARRAY INDEX: DATA-BLOCK INDEX
C          (I*4)  ITT      = ARRAY INDEX: COLLISION ENERGY INDEX
C          (I*4)  NENUM    = NUMBER OF COLLISION ENERGIES FOR CURRENT
C                            DATA-BLOCK
C          (I*4)  IABT     = RETURN CODE FROM 'I4FCTN'
C          (I*4)  IPOS1    = GENERAL USE STRING INDEX VARIABLE
C          (I*4)  IPOS2    = GENERAL USE STRING INDEX VARIABLE
C
C          (R*8)  R8FCTN   = FUNCTION - (SEE ROUTINES SECTION BELOW)
C
C          (L*4)  LBEND    = IDENTIFIES WHETHER THE LAST OF THE  INPUT
C                            DATA SUB-BLOCKS HAS BEEN LOCATED.
C                            (.TRUE. => END OF SUB-BLOCKS REACHED)
C
C          (C*1)  CSLASH   = '/' - DELIMITER FOR 'XXHKEY'
C          (C*2)  C2       = GENERAL USE TWO BYTE CHARACTER STRING
C          (C*3)  CKEY1    = 'FST'    - INPUT BLOCK HEADER KEY
C          (C*4)  CKEY2    = 'TYPE'   - INPUT BLOCK HEADER KEY
C          (C*5)  CKEY3    = 'ALPH0'  - INPUT BLOCK HEADER KEY
C          (C*4)  CKEY4    = 'ISEL'  - INPUT BLOCK HEADER KEY
C          (C*10) C10      = GENERAL USE TEN BYTE CHARACTER STRING
C          (C*80) C80      = GENERAL USE 80 BYTE  CHARACTER  STRING  FOR
C                            THE INPUT OF DATA-SET RECORDS.
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          XXHKEY     ADAS      OBTAIN KEY/RESPONSE STRINGS FROM TEXT
```

```
C          I4EIZ0     ADAS      RETURNS Z0 FOR GIVEN ELEMENT SYMBOL
C          I4FCTN     ADAS      CONVERT CHARACTER STRING TO INTEGER
C          I4UNIT     ADAS      FETCH UNIT NUMBER FOR OUTPUT OF MESSAGES
C          R8FCTN     ADAS      CONVERT CHARACTER STRING TO REAL*8
C
C
C-----------------------------------------------------------------------
C
C NOTES: Based on e9data.for (H P Summers, 06/06/96).
C
C
C VERSION  : 1.1
C DATE     : 27-03-2006
C MODIFIED : Martin O'Mullane
C              - First version
C
C-----------------------------------------------------------------------
      CHARACTER*9         CDONOR(NSTORE)
      CHARACTER*10        CFSTAT(NSTORE)
      CHARACTER*9         CRECVR(NSTORE)
      CHARACTER*2         CTYPE(NSTORE)
      CHARACTER*44        DSNAME
      CHARACTER*2         ESYM
      INTEGER             IEA(NSTORE), ISELA(NSTORE),            IUNIT
      INTEGER             IZ,          IZ0,         IZ1,         NBSEL
      INTEGER             NEDIM,       NSTORE
      REAL*8              ALPH0(NSTORE),            SCX(NEDIM,NSTORE)
      REAL*8              TEEA(NEDIM,NSTORE)
```