ADAS Subroutine e9data

```
C
      SUBROUTINE E9DATA( IUNIT  , DSNAME ,
     &                   NSTORE , NEDIM  ,
     &                   ESYM   , IZ0    ,
     &                   NBSEL  , ISELA  ,
     &                   IZ     , IZ1    ,
     &                   CDONOR , CRECVR , CFSTAT , CTYPE,
     &                   ALPH0  ,
     &                   IEA    ,
     &                   TEEA   , SCX
     &                 )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: E9DATA *********************
C
C  PURPOSE:  TO  FETCH  DATA  FROM  INPUT CHARGE EXCHANGE CROSS-SECTION
C            DATA FOR  GIVEN DONOR AND RECEIVER IONS.
C
C            (MEMBER STORED IN SCX#<DONOR>.DATA (<PREFIX>.#<RECEIVER>)
C             WHERE <DONOR> ='H0' ETC. ; <RECEIVER> ='C2', 'C6' ETC.
C             AND <PREFIX. = <BLANK. OR THREE CHARACTERS
C
C  CALLING PROGRAM: ADAS509/SSCX
C
C  DATA:
C
C           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   = MVS DATA SET NAME 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      INTEGER*4 FUNCTION    -
C                               RETURNS Z0 FOR GIVEN ELEMENT SYMBOL
C          I4FCTN     ADAS      INTEGER*4 FUNCTION    -
C                               CONVERT CHARACTER STRING TO INTEGER
C          I4UNIT     ADAS      INTEGER*4 FUNCTION    -
C                               FETCH UNIT NUMBER FOR OUTPUT OF MESSAGES
C          R8FCTN     ADAS      REAL*8 FUNCTION       -
C                               CONVERT CHARACTER STRING TO REAL*8
C
C AUTHOR:  H. P. SUMMERS, UNIVERSITY OF STRATHCLYDE
C          JA8.08
C          TEL. 0141-553-4196
C
C DATE:    06/06/96
C
C UNIX-IDL PORT: H.P.SUMMERS
C
C VERSION: 1.1 DATE: 30-04-96
C MODIFIED: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C     - PUT UNDER SCCS CONTROL
C
C VERSION: 1.2 DATE: 20-07-07
C MODIFIED: Allan Whiteford
C     - Small modification to comments to allow for automatic
C             documentation preparation.
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)
```