

ADAS Subroutine e1data

```
      SUBROUTINE E1DATA( IUNIT , DSNAME ,
&                        NSTORE , NTDIM , NDDIM ,
&                        IZ0 , IZ , IZ1 , ESYM ,
&                        NBSEL , ISELA ,
&                        CWAVEL , CFILE , CPCODE , CINDM ,
&                        ITA , IDA ,
&                        TETA , TEDA ,
&                        SXB
&                        )
```

```
C-----
C
C ***** FORTRAN77 SUBROUTINE: E1DATA *****
C
C PURPOSE: TO FETCH DATA FROM INPUT IONIZATIONS PER PHOTON FILE
C           FOR A GIVEN EMITTING ION (ELEMENT AND CHARGE).
C           (MEMBER STORED IN IONELEC.DATA - MEMBER PREFIX 'SXB#').
C
C CALLING PROGRAM: ADAS501/SSXB
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 IONIZATIONS
C           PER PHOTON VALUES FOR GIVEN TEMP./DENSITY COMBINATION. EACH
C           DATA-BLOCK IS ANALYSED INDEPENDENTLY OF ANY OTHER DATA-
C           BLOCK.
C
C           THE UNITS USED IN THE DATA FILE ARE TAKEN AS FOLLOWS:
C
C           TEMPERATURES      : EV
C           DENSITIES         : CM-3
C
C SUBROUTINE:
C
C INPUT : (I*4)  IUNIT      = UNIT TO WHICH INPUT FILE IS ALLOCATED.
C INPUT : (C*80) DSNAME    = NAME OF DATA FILE INCLUDING PATH
C
C INPUT : (I*4)  NSTORE    = MAXIMUM NUMBER OF INPUT DATA-BLOCKS THAT
C                           CAN BE STORED.
C INPUT : (I*4)  NTDIM     = MAX NUMBER OF ELECTRON TEMPERATURES ALLOWED
C INPUT : (I*4)  NDDIM     = MAX NUMBER OF ELECTRON DENSITIES ALLOWED
C
C OUTPUT: (I*4)  IZ0       = READ - EMITTING ION - NUCLEAR CHARGE
C OUTPUT: (I*4)  IZ        = READ - EMITTING ION - CHARGE
C OUTPUT: (I*4)  IZ1       = READ - EMITTING ION - CHARGE + 1
C OUTPUT: (C*2)  ESYM      = READ - EMITTING ION - ELEMENT SYMBOL
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: (C*10) CWAVEL() = READ - WAVELENGTH (ANGSTROMS)
```

```

C          DIMENSION: DATA-BLOCK INDEX
C OUTPUT: (C*8)  CFILE() = READ - SPECIFIC ION FILE SOURCE
C          DIMENSION: DATA-BLOCK INDEX
C OUTPUT: (C*8)  CPCODE() = READ - SPECIFIC ION PROCESSING CODE
C          DIMENSION: DATA-BLOCK INDEX
C OUTPUT: (C*2)  CINDM() = READ - METASTABLE INDEX
C          DIMENSION: DATA-BLOCK INDEX
C
C OUTPUT: (I*4)  ITA() = READ - NUMBER OF ELECTRON TEMPERATURES
C          DIMENSION: DATA-BLOCK INDEX
C OUTPUT: (I*4)  IDA() = READ - NUMBER OF ELECTRON DENSITIES
C          DIMENSION: DATA-BLOCK INDEX
C
C OUTPUT: (R*8)  TETA(,) = READ - ELECTRON TEMPERATURES (UNITS: eV)
C          1st DIMENSION: ELECTRON TEMPERATURE INDEX
C          2nd DIMENSION: DATA-BLOCK INDEX
C OUTPUT: (R*8)  TEDA(,) = READ - ELECTRON DENSITIES (UNITS: CM-3)
C          1st DIMENSION: ELECTRON DENSITY INDEX
C          2nd DIMENSION: DATA-BLOCK INDEX
C
C OUTPUT: (R*8)  SXB(,,) =READ - FULL SET OF IONIZATIONS PER PHOTON
C          VALUES.
C          1st DIMENSION: ELECTRON TEMPERATURE INDEX
C          2nd DIMENSION: ELECTRON DENSITY INDEX
C          3rd 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: ELECTRON TEMPERATURE INDEX
C          (I*4)  ITD   = ARRAY INDEX: ELECTRON DENSITY INDEX
C          (I*4)  NTNUM = NUMBER OF ELECTRON TEMPERATURES FOR CURRENT
C          DATA-BLOCK
C          (I*4)  NDNUM = NUMBER OF ELECTRON DENSITIES 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          (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*5)  IONNAM = EMITTING ION READ FROM DATASET
C          (C*6)  CKEY1  = 'FILMEM' - INPUT BLOCK HEADER KEY
C          (C*4)  CKEY2  = 'CODE ' - INPUT BLOCK HEADER KEY
C          (C*4)  CKEY3  = 'INDM ' - INPUT BLOCK HEADER KEY
C          (C*4)  CKEY4  = 'ISEL ' - INPUT BLOCK HEADER KEY
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

C AUTHOR: PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C K1/0/37
C JET EXT. 4569
C

C DATE: 30/04/91
C

C UPDATE: 05/12/91 - PE BRIDEN: IONNAM NOW ALLOWED TO OCCUPY EITHER
C 4 OR 5 SPACES IN THE HEADER.
C

C UPDATE: 23/04/93 - PE BRIDEN - ADAS91: ADDED I4UNIT FUNCTION TO WRITE
C STATEMENTS FOR SCREEN MESSAGES
C

C UPDATE: 24/05/93 - PE BRIDEN - ADAS91: CHANGED I4UNIT(0)-> I4UNIT(-1)
C

C UPDATE: 22/11/94 - L. JALOTA - MODIFIED DSNAME LENGTH FOR UNIX
C

C-----
CHARACTER*8 CFILE (NSTORE)
CHARACTER*2 CINDM (NSTORE)
CHARACTER*8 CPCODE (NSTORE)
CHARACTER*10 CWAVEL (NSTORE)
CHARACTER*80 DSNAME
CHARACTER*2 ESYM
INTEGER IDA (NSTORE) , ISELA (NSTORE)
INTEGER ITA (NSTORE) , IUNIT , IZ , IZ0
INTEGER IZ1 , NBSEL , NDDIM , NSTORE
INTEGER NTDIM
REAL*8 SXB (NTDIM, NDDIM, NSTORE) , TEDA (NDDIM, NSTORE)
REAL*8 TETA (NTDIM, NSTORE)