ADAS Subroutine e4data

```
       SUBROUTINE E4DATA( IUNIT  , DSNAME ,
     &                    NSTORE , NTDIM  ,
     &                    ESYM   , IZ0    ,
     &                    NBSEL  , ISELA  ,
     &                    IZ     , IZ1    ,
     &                    CIION  , CITYPE , CIINFO ,
     &                    ITA    ,
     &                    TETA   , PZD
     &                  )
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: E4DATA *********************
C
C  PURPOSE:  TO  FETCH  DATA  FROM  INPUT RADIATED POWER COEFFICIENTS
C             OF AN ELEMENT AND ITS IONS.
C            (MEMBER STORED IN IONELEC.DATA - MEMBER PREFIX 'PZD#').
C
C  CALLING PROGRAM: ADAS504/SPZD
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  RADIATED
C           POWER COEFFICIENT VALUES FOR GIVEN TEMPERATURES.
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           TEMPERATURES        : EV
C           RATE COEFFT         : W 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 SET BEING READ
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
C  OUTPUT: (C*2)  ESYM     = READ - IONISING ION - ELEMENT SYMBOL
C  OUTPUT: (I*4)  IZ0      = READ - IONISING 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 - RADIATING ION CHARGE
C                                   ( SET TO -1 IF WHOLE ELEMENT)
C                            DIMENSION: DATA-BLOCK INDEX
C  OUTPUT: (I*4)  IZ1()    = READ - RADIATING ION CHARGE +1
```

```
C                                ( SET TO 1 IF WHOLE ELEMENT)
C                              DIMENSION: DATA-BLOCK INDEX
C
C  OUTPUT: (C*5)  CIION()  = READ - RADIATING ION (AS <ESYM>+(IZ()> )
C                              DIMENSION: DATA-BLOCK INDEX
C  OUTPUT: (C*5)  CITYPE() = READ - RADIATION TYPE
C                              DIMENSION: DATA-BLOCK INDEX
C  OUTPUT: (C*20) CIINFO() = READ - INFORMATION STRING
C                              DIMENSION: DATA-BLOCK INDEX
C
C  OUTPUT: (I*4)  ITA()    = READ - NUMBER OF ELECTRON TEMPERATURES
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
C  OUTPUT: (R*8)  PZD(,)    =READ - FULL SET OF ZERO DENSITY RADIATED
C                                    POWER COEFFTS. (W CM**3)
C                              1st DIMENSION: ELECTRON TEMPERATURE 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 ROUTINES SECTION BELOW)
C          (I*4)  IBLK     = ARRAY INDEX: DATA-BLOCK INDEX
C          (I*4)  ITT      = ARRAY INDEX: ELECTRON TEMPERATURE INDEX
C          (I*4)  NTNUM    = NUMBER OF ELECTRON TEMPERATURES 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
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*4)  CKEY1    = '****'  - INPUT BLOCK HEADER KEY
C          (C*5)  CKEY2    = 'TYPE'  - INPUT BLOCK HEADER KEY
C          (C*5)  CKEY3    = 'INFO'  - 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
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/37
C          JET EXT. 2520
C
C DATE:    07/06/91
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:   4/11/94 - L. JALOTA - INCREASED SIZE OF DSNAME TO RUN UNDER
C           UNIX
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 17-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST VERSION
C
C VERSION: 1.2                          DATE: 17-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)/HUGH SUMMERS
C               - MINOR MODIFICATIONS TO FORMAT STATEMENTS
C
C-----------------------------------------------------------------------
      CHARACTER*20        CIINFO(NSTORE)
      CHARACTER*5         CIION(NSTORE),            CITYPE(NSTORE)
      CHARACTER*80        DSNAME
      CHARACTER*2         ESYM
      INTEGER             ISELA(NSTORE),            ITA(NSTORE), IUNIT
      INTEGER             IZ(NSTORE),  IZ0,         IZ1(NSTORE), NBSEL
      INTEGER             NSTORE,      NTDIM
      REAL*8              PZD(NTDIM,NSTORE),        TETA(NTDIM,NSTORE)
```