ADAS Subroutine baubnd

```
       SUBROUTINE BAUBND( ITRAN  , ITRANB , ITRAN2 , I1A   , I2A   ,
     &                    I1BA   , I2BA   , I1A2   , I2A2  , AVAL  ,
     &                    AVALB  , SCOM   , SCOMB  , SCOM2 , NV2   ,
     &                    NDTRN  , NVMAX  , TCODE  , TCODEB, TCODE2,
     &                    INDBL  , NJLEVX , SCOMU  , TCODEU, I1UA  ,
     &                    I2UA   , PRERAT , ILA2   , ISA2  , XJA2  ,
     &                    NDLEV  , IL2    , BNDLS  , NCHK  , IUA   ,
     &                    ILUA   , ISUA   , CSTRGUA, WUA   , XJUA  ,
     &                    IA     , ILA    , ISA    , CSTRGS, WA    ,
     &                    XJA    , IA2    , CSTRGA2, WA2   , NBLEVX,
     &                    INDUL  , NCHKU  , ISORT  , INDBS , AVALU ,
     &                    AVAL2  , PREA   , ITRANU , IUL   , XLSA  ,
     &                    BNDPR  , NBCPRT , IL3    , CPRTAU, IA3   ,
     &                    ILA3   , ISA3   , XJA3   , WA3   , BWNO2 ,
     &                    NPL2   , BWNOA2 , PRTWTA2, CPRTA2, NDMET ,
     &                    IPMDFLG, BWNOAU , CPLA2  , NPLA2 , IPLA2 ,
     &                    ZPLA2  , CPLAU  , NPLAU  , IPLAU , ZPLAU ,
     &                    IMRK   , PRTWTAU, IRCHK  , NZEROS)
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: BAUBND *********************
C
C  PURPOSE: TO UNBUNDLE A SPECIFIC ION FILE ACCORDING TO THE SPLIT UP
C           FRACTIONS OBTAINED FROM A SUPERSTRUCTURE FILE, FILLING IN
C           WITH THE STATISTICAL METHOD OF SARAPH, SEATON & SHEMMING
C           (1969), WHEN NO DATA IS AVAILABLE.
C
C  CALLING PROGRAM: ADAS210
C
C  SUBROUTINE:
C
C  INPUT:
C            (I*4)  NDLEV   = MAXIMUM NUMBER OF LEVELS THAT CAN BE READ
C            (I*4)  NDTRN   = MAX. NUMBER OF TRANSITIONS THAT CAN BE READ
C            (I*4)  NDMET   = MAX. NUMBER OF METASTABLES ALLOWED
C            (I*4)  NVMAX   = MAX. NUMBER OF TEMPERATURES
C            (I*4)  ITRAN   = SUPERSTRUCTURE FILE: NO. OF TRANSITIONS
C            (I*4)  ITRANB  = BUNDLED SUPERSTRUCTURE FILE: NO. OF TRANSITIONS
C            (I*4)  ITRAN2  = INPUT DATA FILE: NO. OF TRANSITIONS
C            (I*4)  I1A()   = TRANSITION: IN SUPERSTRUCTURE FILE
C                             LOWER ENERGY LEVEL INDEX (CASE ' ' & 'P')
C                             SIGNED PARENT INDEX (CASE 'H','R' & 'I')
C            (I*4)  I2A()   = TRANSITION: IN SUPERSTRUCTURE FILE
C                             UPPER ENERGY LEVEL INDEX (CASE ' ' & 'P')
C                             CAPTURING LEVEL INDEX (CASE 'H','R' & 'I')
C            (I*4)  I1BA()  = TRANSITION: IN BUNDLED SUPERSTRUCTURE FILE
C                             LOWER ENERGY LEVEL INDEX (CASE ' ' & 'P')
C                             SIGNED PARENT INDEX (CASE 'H','R' & 'I')
C            (I*4)  I2BA()  = TRANSITION: IN BUNDLED SUPERSTRUCTURE FILE
C                             UPPER ENERGY LEVEL INDEX (CASE ' ' & 'P')
C                             CAPTURING LEVEL INDEX (CASE 'H','R' & 'I')
```

```
C           (I*4)  I1A2()  = TRANSITION: IN INPUT DATA FILE
C                            LOWER ENERGY LEVEL INDEX (CASE ' ' & 'P')
C                            SIGNED PARENT INDEX (CASE 'H','R' & 'I')
C           (I*4)  I2A2()  = TRANSITION: IN INPUT DATA FILE
C                            UPPER ENERGY LEVEL INDEX (CASE ' ' & 'P')
C                            CAPTURING LEVEL INDEX (CASE 'H','R' & 'I')
C           (R*8)  AVAL()  = TRANSITION: IN SUPERSTRUCTURE FILE
C                            A-VALUE (SEC-1)          (CASE ' ')
C                            NEUTRAL BEAM ENERGY      (CASE 'H')
C                            NOT USED             (CASE 'P','R' & 'I')
C           (R*8)  AVALB() = TRANSITION: IN BUNDLED SUPERSTRUCTURE FILE
C                            A-VALUE (SEC-1)          (CASE ' ')
C                            NEUTRAL BEAM ENERGY      (CASE 'H')
C                            NOT USED             (CASE 'P','R' & 'I')
C           (R*8)  AVAL2() = TRANSITION: IN INPUT DATA FILE
C                            A-VALUE (SEC-1)          (CASE ' ')
C                            NEUTRAL BEAM ENERGY      (CASE 'H')
C                            NOT USED             (CASE 'P','R' & 'I')
C           (R*8)  SCOM(,) = TRANSITION: IN SUPERSTRUCTURE FILE
C                            GAMMA VALUES             (CASE ' ' & 'P')
C                            RATE COEFFT.(CM3 SEC-1)(CASE 'H','R' & 'I')
C                           1ST DIMENSION - TEMPERATURE 'SCEF()'
C                           2ND DIMENSION - TRANSITION NUMBER
C           (R*8)  SCOMB(,)= TRANSITION: IN BUNDLED SUPERSTRUCTURE FILE
C                            GAMMA VALUES             (CASE ' ' & 'P')
C                            RATE COEFFT.(CM3 SEC-1)(CASE 'H','R' & 'I')
C                           1ST DIMENSION - TEMPERATURE 'SCEF()'
C                           2ND DIMENSION - TRANSITION NUMBER
C           (R*8)  SCOM2(,)= TRANSITION: IN INPUT DATA FILE
C                            GAMMA VALUES             (CASE ' ' & 'P')
C                            RATE COEFFT.(CM3 SEC-1)(CASE 'H','R' & 'I')
C                           1ST DIMENSION - TEMPERATURE 'SCEF()'
C                           2ND DIMENSION - TRANSITION NUMBER
C           (C*1)  TCODE() = TRANSITION: DATA TYPE POINTER:
C                           ' ' => Electron Impact   Transition
C                           'P' => Proton   Impact   Transition
C                           'H' => Charge   Exchange Recombination
C                           'R' => Free     Electron Recombination
C                           'I' => Coll. ionisation from lower stage ion
C                            IN SUPERSTRUCTURE FILE
C           (C*1)  TCODEB()= TRANSITION: DATA TYPE POINTER:
C                            IN BUNDLED SUPERSTRUCTURE FILE - SAME CODES
C                            AS TCODE ABOVE
C           (C*1)  TCODE2()= TRANSITION: DATA TYPE POINTER:
C                            IN INPUT DATA FILE - SAME CODES
C                            AS TCODE ABOVE
C           (I*4)  NV2     = INPUT DATA FILE: NUMBER OF GAMMA/TEMPERATURE
C                            PAIRS FOR A GIVEN TRANSITION.
C           (I*4)  INDBL() = VECTOR CONTAINING THE BUNDLED SUPERSTRUCTURE
C                            FILE INDICES AT THE ORIGINAL INDEX LOCATIONS
C           (I*4)  NJLEVX  = THE NO. OF LEVELS IN THE SUPERSTRUCTURE FILE
C           (I*4)  IA()    = SUPERSTRUCTURE FILE ENERGY LEVEL INDEX NUMBER
C           (C*18) CSTRGS()= NOMENCLATURE/CONFIGURATION FOR LEVEL 'IA()'
```

```
C                              CONVERTED TO STANDARD FROM EISNER FORM
C             (I*4)  ISA()   = MULTIPLICITY FOR LEVEL 'IA()'
C                              NOTE: (ISA-1)/2 = QUANTUM NUMBER (S)
C             (I*4)  ILA()   = QUANTUM NUMBER (L) FOR LEVEL 'IA()'
C             (R*8)  XJA()   = QUANTUM NUMBER (J-VALUE) FOR LEVEL 'IA()'
C                              NOTE: (2*XJA)+1 = STATISTICAL WEIGHT
C             (R*8)  WA()    = ENERGY RELATIVE TO LEVEL 1 (CM-1) FOR LEVEL
C                             'IA()'
C             (I*4)  IL2     = INPUT DATA FILE: NUMBER OF ENERGY LEVELS
C             (I*4)  IA2()   = INPUT DATA ENERGY LEVEL INDEX NUMBER
C             (C*18) CSTRGA2()= NOMENCLATURE/CONFIGURATION FOR LEVEL 'IA2()'
C             (I*4)  ISA2()  = MULTIPLICITY FOR LEVEL 'IA2()'
C                              NOTE: (ISA2-1)/2 = QUANTUM NUMBER (S)
C             (I*4)  ILA2()  = QUANTUM NUMBER (L) FOR LEVEL 'IA2()'
C             (R*8)  XJA2()  = QUANTUM NUMBER (J-VALUE) FOR LEVEL 'IA2()'
C                              NOTE: (2*XJA2)+1 = STATISTICAL WEIGHT
C             (R*8)  WA2()   = ENERGY RELATIVE TO LEVEL 1 (CM-1) FOR LEVEL
C                             'IA2()'
C             (I*4)  IL3     = INPUT DATA FILE: NUMBER OF ENERGY LEVELS
C             (I*4)  IA3()   = PARENT SUPERSTRUCTURE ENERGY LEVEL INDEX
C                              NUMBER
C             (C*18) CPRTAU()= NOMENCLATURE/CONFIGURATION FOR NEW PARENTS
C             (I*4)  ISA3()  = MULTIPLICITY FOR LEVEL 'IA3()'
C                              NOTE: (ISA3-1)/2 = QUANTUM NUMBER (S)
C             (I*4)  ILA3()  = QUANTUM NUMBER (L) FOR LEVEL 'IA3()'
C             (R*8)  XJA3()  = QUANTUM NUMBER (J-VALUE) FOR LEVEL 'IA3()'
C                              NOTE: (2*XJA3)+1 = STATISTICAL WEIGHT
C             (R*8)  WA3()   = ENERGY RELATIVE TO LEVEL 1 (CM-1) FOR LEVEL
C                             'IA3()'
C             (I*4)  BNDLS() = LEVEL/TERM SELECTION VECTOR
C             (I*4)  BNDPR() = PARENT METASTABLE SELECTION VECTOR
C             (I*4)  NBLEVX  = THE NO. OF LEVELS IN THE BUNDLED
C                              SUPERSTRUCTURE FILE
C             (R*8)  XLSA()  = QUANTUM NUMBER (J-VALUE) FOR BUNDLED
C                              SUPERSTRUCTURE LEVEL 'I2BA()'
C                              NOTE: (2*XLSA)+1 = STATISTICAL WEIGHT
C             (I*4)  NBCPRT  = NUMBER OF SELECTED CONTRIBUTIONS TO PARENTS
C             (R*8)  BWNO2   = IONISATION POTENTIAL (CM-1) OF LOWEST PARENT
C                              IN INPUT DATA FILE
C             (I*4)  NPL2    = NUMBER OF PARENTS ON FIRST LINE OF INPUT
C                              DATA FILE AND USED IN LEVEL ASSIGNMENTS
C             (R*8)  BWNOA2()= IONISATION POTENTIAL (CM-1) OF PARENTS
C                              IN INPUT DATA FILE
C             (R*8)  PRTWTA2()= PARENT WEIGHT FOR BWNOA2()
C             (C*9)  CPRTA2()= PARENT NAME IN BRACKETS IN INPUT DATA FILE
C             (C*1)  CPLA2() = CHAR. SPECIFYING 1ST PARENT FOR LEVEL 'IA2()'
C                                 INTEGER - PARENT IN BWNOA2() LIST
C                                 'BLANK' - PARENT BWNOA2(1)
C                                   'X'   - DO NOT ASSIGN A PARENT
C             (I*4)  NPLA2() = NO. OF PARENT/ZETA CONTRIBUTIONS TO IONIS.
C                              OF LEVEL IN INPUT DATA FILE
C             (I*4)  IPLA2(,)= PARENT INDEX FOR CONTRIBUTIONS TO IONIS.
C                              OF LEVEL IN INPUT DATA FILE
```

```
C                              1ST DIMENSION: PARENT INDEX
C                              2ND DIMENSION: LEVEL INDEX
C            (I*4)  ZPLA2(,)= EFF. ZETA PARAM. FOR CONTRIBUTIONS TO IONIS.
C                              OF LEVEL IN INPUT DATA FILE
C                              1ST DIMENSION: PARENT INDEX
C                              2ND DIMENSION: LEVEL INDEX
C            (I*4)  IPMDFLG = FLAG FOR PARENT SUPERSTRUCTURE FILE
C                              AVAILABILITY
C
C  OUTPUT:
C            (I*4)  ITRANU  = OUTPUT DATA FILE: NO. OF TRANSITIONS
C            (I*4)  I1UA()  = TRANSITION: IN OUTPUT DATA FILE
C                              LOWER ENERGY LEVEL INDEX (CASE ' ' & 'P')
C                              SIGNED PARENT INDEX (CASE 'H','R' & 'I')
C            (I*4)  I2UA()  = TRANSITION: IN OUTPUT DATA FILE
C                              UPPER ENERGY LEVEL INDEX (CASE ' ' & 'P')
C                              CAPTURING LEVEL INDEX (CASE 'H','R' & 'I')
C            (R*8)  AVALU() = TRANSITION: IN OUTPUT DATA FILE
C                              A-VALUE (SEC-1)          (CASE ' ')
C                              NEUTRAL BEAM ENERGY      (CASE 'H')
C                              NOT USED             (CASE 'P','R' & 'I')
C            (R*8)  SCOMU(,)= TRANSITION: IN OUTPUT DATA FILE
C                              GAMMA VALUES             (CASE ' ' & 'P')
C                              RATE COEFFT.(CM3 SEC-1)(CASE 'H','R' & 'I')
C                             1ST DIMENSION - TEMPERATURE 'SCEF()'
C                             2ND DIMENSION - TRANSITION NUMBER
C            (C*1)  TCODEU()= TRANSITION: DATA TYPE POINTER:
C                              IN OUTPUT DATA FILE - SAME CODES
C                              AS TCODE ABOVE
C            (R*8)  PRERAT(,)=ARRAY OF PREMULTIPLIERS FOR THE J-RESOLVED
C                              LEVELS. RATIO OF INPUT DATA TO BUNDLED DATA
C                              FOR A TRANSITION
C                             1ST DIMENSION - TEMPERATURE 'SCEF()'
C                             2ND DIMENSION - TRANSITION NUMBER
C            (R*8)  PREA()  = PREMULTIPLIERS FOR THE SUPERSTRUCTURE
C                              A-VALUES. RATIO OF STAT. WEIGHTED INPUT
C                              DATA TO BUNDLED DATA.
C            (I*4)  IUA()   = ENERGY LEVEL INDEX NUMBER
C            (C*18) CSTRGUA()= NOMENCLATURE/CONFIGURATION FOR LEVEL 'IUA()'
C            (I*4)  ISUA()  = MULTIPLICITY FOR LEVEL 'IUA()'
C                              NOTE: (ISUA-1)/2 = QUANTUM NUMBER (S)
C            (I*4)  ILUA()  = QUANTUM NUMBER (L) FOR LEVEL 'IUA()'
C            (R*8)  XJUA()  = QUANTUM NUMBER (J-VALUE) FOR LEVEL 'IUA()'
C                              NOTE: (2*XJUA)+1 = STATISTICAL WEIGHT
C            (R*8)  WUA()   = ENERGY RELATIVE TO LEVEL 1 (CM-1) FOR LEVEL
C                             'IUA()'
C            (I*4)  INDUL() = VECTOR CONTAINING THE UNBUNDLED FILE
C                              INDICES AT THE ORIGINAL INDEX LOCATIONS
C            (I*4)  IUL     = OUTPUT DATA FILE: NUMBER OF ENERGY LEVELS
C            (R*8)  BWNOAU()= IONISATION POTENTIAL (CM-1) OF PARENTS
C                              IN OUTPUT DATA FILE
C            (R*8)  PRTWTAU()= PARENT WEIGHT FOR BWNOAU()
C            (C*1)  CPLAU() = CHAR. SPECIFYING 1ST PARENT FOR LEVEL 'IUA()'
```

```
C                                INTEGER - PARENT IN BWNOA2() LIST
C                                'BLANK' - PARENT BWNOA2(1)
C                                  'X'   - DO NOT ASSIGN A PARENT
C            (I*4)  NPLAU() = NO. OF PARENT/ZETA CONTRIBUTIONS TO IONIS.
C                             OF LEVEL IN OUTPUT DATA FILE
C            (I*4)  IPLAU(,)= PARENT INDEX FOR CONTRIBUTIONS TO IONIS.
C                             OF LEVEL IN OUTPUT DATA FILE
C                            1ST DIMENSION: PARENT INDEX
C                            2ND DIMENSION: LEVEL INDEX
C            (I*4)  ZPLAU(,)= EFF. ZETA PARAM. FOR CONTRIBUTIONS TO IONIS.
C                             OF LEVEL IN OUTPUT DATA FILE
C                            1ST DIMENSION: PARENT INDEX
C                            2ND DIMENSION: LEVEL INDEX
C            (I*4)  NCHK()  = VECTOR NOTING REPEATED USER SELECTIONS
C                             AND HOW OFTEN THEY OCCUR, FOR LEVELS
C            (I*4)  NCHKU() = VECTOR NOTING THE LEVELS SELECTED FOR
C                             UNBUNDLING AND THEIR NEW POSITIONING
C            (I*4)  ISORT() = CROSS REFERENCE VECTOR FOR NEW INDEXING
C            (I*4)  INDBS() = CROSS REFERENCE VECTOR FOR NEW BUNDLED
C                             SUPERSTRUCTURE INDEXING
C            (I*4)  IMRK()  = VECTOR NOTING REPEATED USER SELECTIONS
C                             AND HOW OFTEN THEY OCCUR, FOR PARENTS
C            (I*4)  IRCHK() = VECTOR NOTING THE PARENTS SELECTED FOR
C                             UNBUNDLING AND THEIR NEW POSITIONING
C
C  ROUTINES: NONE
C
C  AUTHOR:   DAVID H.BROOKS (UNIV.OF STRATHCLYDE) EXT.4213/4205
C
C  DATE:     12/01/96
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 22-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - PUT UNDER SCCS CONTROL
C
C VERSION: 1.2                          DATE: 26-1-96
C MODIFIED: DAVID H.BROOKS
C               - ALTERED MATCHING OF LSJ LEVELS TO DATASETS IN ORDER
C                 TO ALLOW PARTIAL SPLITTING OF THE SOURCE FILE.
C
C VERSION: 1.3                          DATE: 19-11-98
C MODIFIED: DAVID H.BROOKS
C               - MODIFIED TO ALLOW EXTRA LEVELS TO BE INTERSPERSED
C                 WITH THE ONES BEING USED. THESE ARE OMITTED FROM
C                 THE ACTUAL CALCULATIONS BY NOTING THEIR POSITIONS
C                 IN BNDLS.
C
C-----------------------------------------------------------------------
      CHARACTER           CPLA2(NDLEV),             CPLAU(NDLEV)
      CHARACTER*9         CPRTA2(NDMET),            CPRTAU(2*NDMET)
      CHARACTER*18        CSTRGA2(NDLEV),           CSTRGS(NDLEV)
```

```
CHARACTER*18        CSTRGUA(NDLEV)
CHARACTER           TCODE(NDTRN),             TCODE2(NDTRN)
CHARACTER           TCODEB(NDTRN),            TCODEU(NDTRN)
INTEGER             BNDLS(NDLEV),             BNDPR(NDLEV)
INTEGER             I1A(NDTRN),  I1A2(NDTRN), I1BA(NDTRN)
INTEGER             I1UA(NDTRN), I2A(NDTRN),  I2A2(NDTRN)
INTEGER             I2BA(NDTRN), I2UA(NDTRN), IA(NDLEV)
INTEGER             IA2(NDLEV),  IA3(NDLEV),  IL2,          IL3
INTEGER             ILA(NDLEV),  ILA2(NDLEV), ILA3(NDLEV)
INTEGER             ILUA(NDLEV), IMRK(2*NDMET)
INTEGER             INDBL(NJLEVX),            INDBS(NDLEV)
INTEGER             INDUL(NDLEV),             IPLA2(NDMET,NDLEV)
INTEGER             IPLAU(2*NDMET,NDLEV),     IPMDFLG
INTEGER             IRCHK(2*NDMET),           ISA(NDLEV)
INTEGER             ISA2(NDLEV), ISA3(NDLEV), ISORT(NDLEV)
INTEGER             ISUA(NDLEV), ITRAN,       ITRAN2,       ITRANB
INTEGER             ITRANU,      IUA(NDLEV),  IUL,          NBCPRT
INTEGER             NBLEVX,      NCHK(NJLEVX)
INTEGER             NCHKU(NDLEV),             NDLEV,        NDMET
INTEGER             NDTRN,       NJLEVX,      NPL2
INTEGER             NPLA2(NDLEV),             NPLAU(NDLEV)
INTEGER             NV2,         NVMAX,       NZEROS
REAL*8              AVAL(NDTRN), AVAL2(NDTRN)
REAL*8              AVALB(NDTRN),             AVALU(NDTRN)
REAL*8              BWNO2,       BWNOA2(NDMET)
REAL*8              BWNOAU(2*NDMET),          PREA(NDTRN)
REAL*8              PRERAT(NVMAX,NDTRN),      PRTWTA2(NDMET)
REAL*8              PRTWTAU(2*NDMET),         SCOM(NVMAX,NDTRN)
REAL*8              SCOM2(NVMAX,NDTRN),       SCOMB(NVMAX,NDTRN)
REAL*8              SCOMU(NVMAX,NDTRN),       WA(NDLEV)
REAL*8              WA2(NDLEV),  WA3(NDLEV),  WUA(NDLEV)
REAL*8              XJA(NDLEV),  XJA2(NDLEV), XJA3(NDLEV)
REAL*8              XJUA(NDLEV), XLSA(NDLEV), ZPLA2(NDMET,NDLEV)
REAL*8              ZPLAU(2*NDMET,NDLEV)
```