

```

/* #####
/* #####          DS.AML          #####
/* #####

/* =====
/* PURPOSE:
/* Animates a time series from Arc Info ASCII grids created by
/* the CASC2D-SED hydrological model.  The user has the possibility
/* of creating an animated MPEG file (movie) from the ArcPlot
/* display window by creating a screenshot of each display.
/* In general, this AML can be used to create a movie from any
/* time series of Arc Info ASCII grids if these ASCII grids have
/* the same Prefix (root) name and the Suffix (file extension) in
/* an ordered sequence of arabic numbers.
/*
/* =====
/* INPUT
/* 1. A time series of Arc Info ASCII grids:
/*   Example 1: one map per frame and n number of frames are shown
/*   depth.1; depth.2; ....; depth.n;
/*   Example 2: four maps per frame and n number of frames are shown
/*   depth.1; depth.2; ....; depth.n;
/*   erosion.1; erosion.2; ....erosion.n;
/*   flow.1; flow.2; ....; flow.n;
/*   rain.1; rain.2; ....; rain.n;
/* 2. A parameter file that will control the display of the grids or
/* map composition.  Optionally, this parameter file might not be
/* given but the user needs to enter data manually.
/*
/* =====
/* OUTPUT:
/* 1. A series of Arc Info grids (if user doesn't opt to kill them)
/*   after making the movie)
/*   Example: User chooses to show every other grid in his input
/*   series, starting in grid 6 and ending in grid 10:
/*   depth6; depth8; depth10
/* 2. An MPEG file (if user decides to create a movie using the
/*   generated grids)
/*
/* =====
/* USAGE:
/* Arc: &run ds <ParameterFile>
/*
/* =====
/* ARGUMENTS:
/* ParamFile: Name of the input data file. If not specified,
/*   the user must enter the input data interactively.
/*
/* =====
/* VARIABLES:
/* Globals:
/* .CreateMovie: Index: 1: Movie is created; 0: not created
/* .ElapsedTime: Elapsed time (secs) between CASC2D generated maps
/* .EndFrame: Frame in which the display or movie will end
/* .FrameStep: Number of frames to skip when displaying series
/* .iter: Counter used to name multiple sequential images
/* .MapTitleXX: Title for each of the displayed maps
/* .MovieName: Name given to the final MPEG file (movie)
/* .NameXX: Prefix (root) given to each of the thematic series

```

```

/* .RMTXX: Remap table for each of the thematic series
/* .SecsPause: Seconds to halt the series display
/* .ShdSetXX: Shade set for each of the thematic series
/* .StartFrame: Frame in which the display or movie will end
/* .wPath: workspace location. Enter full path if the workspace
/*          is in another location
/*
/* =====
/* CALLS:
/*   APdisplay.aml;
/*
/* =====
/* ROUTINES:
/*   GetUserInputData -- Prompts the user to enter input data
/*   GetFileInputData -- Input data is read from a file
/*   CreateGrids      --
/*       1. Creates the grids from the ArcInfo
/*          ASCII files if they don't exists.
/*          They will be named as: PrefixSufix
/*          Example: depth1; depth2; ....depthn;
/*       2. Writes grid stats. in text files
/*   DisplayGrids     -- Displays the series of grids in an Arc Plot
/*          display window according to the specified
/*          remap table and shadesets defined previously
/*          by the user.
/*   Animate          -- Executes the MPEG encoder
/*   KillAllGrids     -- Created grids can be cleaned once the screen
/*          shot has been taken if the user decides so.
/*   GetUserSpecs     -- Get the user's ArcPlot configuration
/*   PutUserSpecs     -- Restores the user's AP configuration
/*   bailout          -- bails of of aml
/* =====
/* NOTES:
/* Must have the Berkeley MPEG encoder in:
/*          $ARCHOME\bin\mpeg_encode.exe
/*
/* =====
/* HISTORY:
/* 9 Aug 1996 Original coding - Ian DeMerchant
/* (as mpeg_encode.aml in ArcTools)
/* 15 Nov 2000 Major revision - Stephen Lead (slead@esriau.com.au)
/* (as movie.aml)
/* 15 Jul 2001 Revision -- Rosalia Rojas (CSU)
/* (as DisplaySeries.aml)
/* This AML has been rewritten for handling different options.
/* The movie can display 1, 2, 3, 4 or 6 maps per frame
/* User is able to define the remap table and color scheme
/* for each of the maps in the frame.
/* User is able to define a starting and ending frame and
/* show them in a predefined interval.
/*
/* =====

```

```
&ARGS DSParamFile
```

```

&IF [NULL %DSParamFile%] &THEN
  &CALL GetUserInputData
&ELSE

```

```

    &CALL GetFileInputData
&CALL CheckInputErrors
&CALL CreateGrids

/* IN CASE THAT A MOVIE IS NOT TO BE CREATED, THE USER IS GIVEN THE
/* POSSIBILITY OF DISPLAYING THE GRIDS AS MANY TIMES AS (S)HE WISHES

&SV ShowSeriesAgain = .FALSE.

&do &until ^ %ShowSeriesAgain%
    &CALL DisplayGrids
    &IF %.CreateMovie% = 0 &THEN
        &DO
            &TYPE
            &TYPE !!! Hello there !!!
            &SV ShowSeriesAgain = ~
            [QUERY 'Do you want to display again the time series' .FALSE.]

            /* Include the next line in case that you changed your mind
            /* after displaying the series (ex. you decide to kill grids)
            &IF %ShowSeriesAgain% &THEN &CALL GetFileInputData
            &TYPE
        &END
    &end

&IF %.CreateMovie% = 1 &THEN ; &CALL Animate

&CALL KillAllGrids

&RETURN

/* #####
/* ##### ROUTINE GET_USER_INPUT_DATA #####
/* #####

&ROUTINE GetUserInputData

/* GET THE NUMBER OF MAPS PER FRAME */

&TYPE
&SV .MapsPerFrame = ~
    [RESPONSE 'How many maps per frame? 1,2,3,4 or 6?']
&TYPE

/* DEFINE WHICH FRAMES WILL BE SHOWN

&SV .StartFrame = [RESPONSE 'Start movie in frame number']
&TYPE

&SV .EndFrame = [RESPONSE 'End movie in frame number']
&TYPE

&SV .ElapsedTime = ~
    [RESPONSE 'Elapsed show time between frames (sec.)']
&TYPE

&SV .FrameStep = [RESPONSE 'Show frame step']
&TYPE

```

```

&SV .SecsPause = [RESPONSE 'Seconds to pause frames display in AP']

/* Ask for the location of the folder where the ASCII grids are
&SV .wPath = [RESPONSE 'ASCII grids location (full path)']
&TYPE

/* GET THE NAMES OF THOSE MAPS

&DO MapNumber = 1 &TO %.MapsPerFrame%
  &sv .Name%MapNumber% = ~
  [RESPONSE 'Name (prefix) of map no. '%MapNumber%' :']
&END
&TYPE

/* ASK THE USER TO ENTER THE REMAP TABLE NAME FOR EACH OF
/* THE MAPS (S)HE WANTS TO DISPLAY

&DO MapNum = 1 &TO %.MapsPerFrame%
  &sv .RMT%MapNum% = ~
  [RESPONSE 'Remap table name for the '[value Name%MapNum%]' map']
&END
&TYPE

/* ASK THE USER TO ENTER THE SHADESET FILE NAME FOR EACH OF
/* THE MAPS (S)HE WANTS TO DISPLAY

&DO MapNo = 1 &TO %.MapsPerFrame%
  &sv .ShdSet%MapNo% = ~
  [RESPONSE 'Shadeset file name for the '[value Name%MapNo%]' map']
&END
&TYPE

/* ASK THE USER TO ENTER THE TITLE TO DISPLAY FOR EACH OF
/* THE MAPS (S)HE WANTS TO DISPLAY

&DO MapNo = 1 &TO %.MapsPerFrame%
  &sv .MapTitle%MapNo% = ~
  [RESPONSE 'Title to display for the '[value Name%MapNo%]' map']
&END
&TYPE

/* ASK WHETHER THE USER WANTS TO CLEAN THE WORKSPACE
/* AFTER THE MOVIE IS DONE.

&SV OrderToKill = ~
  [RESPONSE 'Kill grids after they are displayed (1:yes;0:no) ?']
&TYPE

/* SCARE THE USER IF (S)HE WANTS TO CREATE THE MPEG FILE
&TYPE
&TYPE WARNING !!!!
&TYPE Be aware that creating a movie will take for about
&TYPE a minute per frame. Make sure that this time series
&TYPE is what you want to be displayed.
&TYPE
&TYPE
&TYPE

```

```

&SV .CreateMovie = ~
  [RESPONSE 'would you like to create a movie (1:yes;0:no) ?']
&TYPE

&IF %CreateMovie% = 1 &THEN
  &SV .MovieName = [RESPONSE 'Movie name']
&TYPE

/*&SV .3Drendering = ~
/*  [RESPONSE 'would you like a 3-D representation (1:yes;0:no) ?']
/*&TYPE

/*&SV .SurfaceModel = ~
/*  [RESPONSE 'Name of the surface model (include path) ?']
/*&TYPE

/*&SV .Zscale = ~
/*  [RESPONSE 'Vertical scale (>1 exaggeration; <1 reduccion):']
/*&TYPE

&RETURN

/* #####
/* #####      ROUTINE GET_FILE_INPUT_DATA      #####
/* #####

&ROUTINE GetFileInputData

&SV AMLunit = [open %DSParamFile% openstat -read]

/* First four lines are comments
&SV line1 = [read %AMLunit% readstat] /* comment line
&SV line2 = [read %AMLunit% readstat] /* comment line
&SV line3 = [read %AMLunit% readstat] /* comment line
&SV line4 = [read %AMLunit% readstat] /* comment line
&TYPE

&SV line5 = [unquote [read %AMLunit% readstat]]
&SV .MapsPerFrame = [extract 2 %line5%]
&TYPE Maps Per Frame: %.MapsPerFrame%
&TYPE

&SV line6 = [unquote [read %AMLunit% readstat]]
&SV .StartFrame = [extract 2 %line6%]
&TYPE Starting Frame = %.StartFrame%
&TYPE

&SV line7 = [unquote [read %AMLunit% readstat]]
&SV .EndFrame = [extract 2 %line7%]
&TYPE Ending Frame = %.EndFrame%
&TYPE

&SV line8 = [unquote [read %AMLunit% readstat]]
&SV .ElapsedTime = [extract 2 %line8%]
&TYPE Elapsed time between frames = %.ElapsedTime%
&TYPE

&SV line9 = [unquote [read %AMLunit% readstat]]

```

```

&SV .FrameStep = [extract 2 %line9%]
&TYPE Frame Step = %.FrameStep%
&TYPE

&SV line10 = [unquote [read %AMLunit% readstat]]
&SV .SecsPause = [extract 2 %line10%]
&TYPE Seconds to pause frame display in AP = %.SecsPause%
&TYPE

&SV line11 = [read %AMLunit% readstat] /* blank line

&SV line12 = [UNQUOTE [READ %AMLunit% readstat] ]
&SV .wPath = [extract 2 %line12%]
&TYPE Time series grids location = %.wPath%
&TYPE

&SV line13 = [unquote [read %AMLunit% readstat]]
&DO MapNumber = 1 &TO %.MapsPerFrame%
  &SV .Name%MapNumber% = ~
    [EXTRACT [CALC %MapNumber% + 1] %line13%]
  &TYPE Map No. %MapNumber%: [VALUE .Name%MapNumber%]
&END
&TYPE

&SV line14 = [unquote [read %AMLunit% readstat]]
&DO MapNumber = 1 &TO %.MapsPerFrame%
  &sv .RMT%MapNumber% = ~
    [EXTRACT [CALC %MapNumber% + 1] %line14%]
  &TYPE Map No. %MapNumber% Remap Table: [VALUE .RMT%MapNumber%]
&END
&TYPE

&SV line15 = [unquote [read %AMLunit% readstat]]
&DO MapNumber = 1 &TO %.MapsPerFrame%
  &sv .ShdSet%MapNumber% = ~
    [EXTRACT [CALC %MapNumber% + 1] %line15%]
  &TYPE Map No. %MapNumber% shadeset: [VALUE .ShdSet%MapNumber%]
&END
&TYPE

&SV line16 = [unquote [read %AMLunit% readstat]]
&DO MapNumber = 1 &TO %.MapsPerFrame%
  &sv .MapTitle%MapNumber% = ~
    [EXTRACT [CALC %MapNumber% + 1] %line16%]
  &TYPE Map No. %MapNumber% title: [VALUE .MapTitle%MapNumber%]
&END
&TYPE

&SV line17 = [unquote [read %AMLunit% readstat]]
&DO MapNumber = 1 &TO %.MapsPerFrame%
  &sv .MapSubTitle%MapNumber% = ~
    [EXTRACT [CALC %MapNumber% + 1] %line17%]
  &TYPE Map No. %MapNumber% subtitle: [VALUE .MapSubTitle%MapNumber%]
&END
&TYPE

&SV line18 = [unquote [read %AMLunit% readstat]]
&SV .OrderToKill = [extract 2 %line18%]
&IF %.OrderToKill% = 1 &THEN; &TYPE Grids deleted after displayed

```

```

&ELSE &TYPE Generated grids will not be deleted
&TYPE

&SV line19 = [unquote [read %AMLunit% readstat]]
&SV .CreateMovie = [extract 2 %line19%]
&IF %.CreateMovie% = 1 &THEN; &TYPE An MPEG file will be created
&ELSE &TYPE No MPEG file will be created
&TYPE

&SV line20 = [unquote [read %AMLunit% readstat]]
&SV .MovieName = [extract 2 %line20%]
&TYPE Movie Name = %.MovieName%
&TYPE

&SV line21 = [read %AMLunit% readstat] /* comment line
&SV line22 = [read %AMLunit% readstat] /* comment line

&SV line23 = [unquote [read %AMLunit% readstat]]
&SV .3Drendering = [extract 2 %line23%]
&IF %.3Drendering% = 1 &THEN; &TYPE 3-D grids will be displayed
&ELSE &TYPE 2-D grids will be displayed
&TYPE

/*&IF %.3Drendering% = 1 &THEN
/* &DO
/* &SV line24 = [unquote [read %AMLunit% readstat]]
/* &SV .SurfaceModel = [extract 2 %line24%]
/* &TYPE Surface Model: %.SurfaceModel%
/* &TYPE
/* &SV line25 = [unquote [read %AMLunit% readstat]]
/* &SV .Zscale = [extract 2 %line25%]
/* &TYPE Z scale: %.Zscale%
/* &TYPE
/* &END

&SV closestat = [CLOSE %AMLunit%]

&RETURN

/* #####
/* #####          ROUTINE CREATE_GRIDS          #####
/* #####

&routine CreateGrids

/* OPEN AND WRITE TITLES IN THE FILES THAT WILL HOLD THE GRIDS STATS.

&DO i = 1 &TO %.MapsPerFrame%
&SV fileexists = [EXIST %wPath%[value .Name%i%]_desc.txt -FILE]
&IF %fileexists% &THEN
&SV NULL [DELETE %wPath%[value .Name%i%]_desc.txt]
&SV fileunit%i% ~
    [open %wPath%[value .Name%i%]_desc.txt openstat -WRITE]
&SV NULL [WRITE [value fileunit%i%] ~
'GridName      Min.          Max.          Mean          StdDev']
&SV NULL [WRITE [value fileunit%i%] ~
'-----']]
&END

```

```

&FORMAT 3

/* THE INPUT ASCII GRIDS (FROM CASC2D_SED) ARE CALLED
/*      ex. depth.1, depth.2, AND SO ON
/* GRIDS ARE GOING TO BE CALLED ex. depth1, depth2 AND SO ON
/* IF OLD GRIDS ARE PRESENT, THEY WILL BE REUSED, OTHERWISE, THEY
/* WILL BE CREATED. MAKE SURE THAT THOSE OLD GRIDS ARE THE ONES YOU
/* WANT TO USE, OTHERWISE, KILL THEM BEFORE STARTING THE CURRENT AML

&DO FrameNumber = %.StartFrame% &TO %.EndFrame% &BY %.FrameStep%

    &TYPE Frame Number: %FrameNumber%

    &DO MapNumber = 1 &TO %.MapsPerFrame%
        &SV String := %.wPath%[value .Name%MapNumber%]
        &SV gridexist = [EXISTS %String%%FrameNumber% -grid]
        &IF %gridexist% eq .FALSE. &THEN
            &DO
                &TYPE Creating grid
                ASCIIGRID ~
                    %String%.%FrameNumber% %String%%FrameNumber% FLOAT
                &TYPE Calculating statistics
                &DESCRIBE %String%%FrameNumber%
                &SV DataLine := ~
                    [FORMAT '%1,10% %2,-15% %3,-15% %4,-15% %5,-15%' ~
                    %GRD$GRID% %GRD$ZMIN% %GRD$ZMAX% %GRD$MEAN% %GRD$STDV%]
                &SV NULL [WRITE [value fileunit%MapNumber%] %DataLine%]
                /*&CALL ComputeStats
            &END
        &END

&END

&END

&DO i = 1 &TO %.MapsPerFrame% /* Closing the stats. files
    &SV NULL [CLOSE [value fileunit%i%]]
&END

&RETURN

/* #####
/* #####          ROUTINE DISPLAY_GRIDS          #####
/* #####

&ROUTINE DisplayGrids

/* THE VARIABLE iter WAS ORIGINALLY INSIDE THE animate.aml
/* WE INITIALIZE IT OUTSIDE THE LOOP THAT CREATES THE GRIDS

&SV .iter -1

AP

&CALL GetUserSpecs

&SV timesec 0 /* Initialize total time to zero before displaying

&FORMAT 1

```



```

/*&IF %.MapsPerFrame% = 1 &THEN
/*  &RUN DS1.AML
/*&ELSE
    &RUN APdisplay.aml

&CALL PutUserSpecs

QUIT

&RETURN

/* #####
/* #####          ROUTINE ANIMATE          #####
/* #####

/* THIS PART OF THE CODE IS A MODIFIED COPY OF THE animate
/* routine WRITEN BY STEPHEN LEAD IN THE animate.aml

&ROUTINE animate

/* TO CREATE THE ANIMATED MPEG FILE, WE USE THE BERKELEY MPEG ENCODER
/* WITH THE DEFAULT MPEG PARAMETER FILE DEFINED IN THIS AML.

/* THE USER CAN CHANGE THESE PARAMETER VALUES TO HIS/HER LIKING
/* ACCORDING TO THE USERS-GUIDE.PS CREATED AT THE BERKELEY MULTIMEDIA
/* RESEARCH CENTER, WHICH CAN BE FOUND AT:
/* http://bmrc.berkeley.edu/frame/research/mpeg/mpeg\_encode.html

/* A GOOD EXPLANATION AND EXAMPLE OF THE MPEG PARAMETER FILE IS FOUND
/* AT: http://www.math.arizona.edu/swig/animation/MPEG/\(by Mark Hays\)

/* Calculate the first and last images - the temporary files
/* contain a prefix with 5 trailing digits. We need to substitute
/* zeros for any leading blank values.

&s first 00000
&s last %.iter%
    &do i = 1 &to [calc 5 - [length %.iter%]]
        &s last 0%last%
    &end

/* Write out a parameter file for this animation.
&s param_file [scratchname -file]
&s fileunit [open %param_file% openstat -write]
&if %openstat% <> 0 &then
    &do
        &s str Could not create %param_file%...; &call bailout
    &end

&s null [write %fileunit% 'PATTERN IBBPBBPBBPBB']
&s null [write %fileunit% 'INPUT']
&setchar &function '|' '*' '|'
&s null |* write %fileunit% ~
            |* quote %.MovieName%* [%first%-%last%] *| *|
&setchar &function [ ]
&s null [write %fileunit% 'END_INPUT']
&s null [write %fileunit% [quote OUTPUT %.wPath%.MovieName%]]

```

```

&s null [write %fileunit% [quote INPUT_DIR [show work]]]
&s null [write %fileunit% ~
  [quote INPUT_CONVERT [joinfile [joinfile ~
    [pathname $ARCHOME] bin -SUB] rasttopnm -FILE] *]]
&s null [write %fileunit% 'BASE_FILE_FORMAT PNM']
&s null [write %fileunit% 'SLICES_PER_FRAME 1']
&s null [write %fileunit% 'PIXEL HALF']
&s null [write %fileunit% 'RANGE 9']
&s null [write %fileunit% 'PSEARCH_ALG TWOLEVEL']
&s null [write %fileunit% 'BSEARCH_ALG CROSS2']
&s null [write %fileunit% 'GOP_SIZE 12']
&s null [write %fileunit% 'PQSCALE 10']
&s null [write %fileunit% 'IQSCALE 10']
&s null [write %fileunit% 'BQSCALE 10']
&s null [write %fileunit% 'REFERENCE_FRAME ORIGINAL']
&s null [write %fileunit% 'FORCE_ENCODE_LAST_FRAME']
&s null [close %fileunit%]

/* EXECUTE THE ENCODER
&sys [joinfile [joinfile [pathname $ARCHOME] bin -SUB] ~
  mpeg_encode -FILE] %param_file%

/* DELETE THE TEMPORARY FILES
&s deletestat [DELETE %param_file% -file]
&DO i = 0 &TO %.iter%
&SV value %i%
  &DO a = 1 &TO [calc 5 - [LENGTH %i%]]
  &S value 0%value%
  &S delstat [DELETE %.MovieName%%value% -file]
  &END
&END

&RETURN

/* #####
/* #####          ROUTINE KILL_ALL_GRIDS          #####
/* #####

/* KILL THE INPUT GRIDS IF THE USER DECIDED SO

&ROUTINE KillAllGrids

&IF %.OrderToKill% = 1 &THEN

  &DO FrameNumber = %.StartFrame% &TO %.EndFrame% &BY %.FrameStep%

  &type Killing Grids Number: %FrameNumber%

  &DO MapNumber = 1 &TO %.MapsPerFrame%
  &sv String := %.wPath%[value .Name%MapNumber%]
  &sv gridexist = [exists %String%%FrameNumber% -grid]
  &if %gridexist% &then
    KILL %String%%FrameNumber% ALL
  &END

&END

&RETURN

```

```

/* #####
/* #####          ROUTINE GET_USER_SPECS          #####
/* #####

/* DETERMINES THE USERS SPECIFICATIONS BEFORE IMPLEMENTING NEW ONES
/* WITH THIS AML

&ROUTINE GetUserSpecs

    /* GET THE NUMBER OF DECIMALS
    &SV UsersDecimals [SHOW &FORMAT]
    &TYPE UsersDecimals %UsersDecimals%

    /* GET THE DISPLAY WINDOW SPECIFICATION
    &SV UsersAPdisplay [SHOW DISPLAY]
    &TYPE UsersAPdisplay %UsersAPdisplay%

    /* GET THE SHADESET
    &SV UsersShadeSet [SHOW SHADESET]
    &TYPE UsersShadeSet %UsersShadeSet%

    /* GET THE TEXT SPECIFICATIONS
    &SV UserTextFont [SHOW TEXTFONT]
    &SV UserTextQuality [SHOW TEXTQUALITY]
    &SV UserTextSpacing [SHOW TEXTSPACING]
    &SV UserTextSize [SHOW TEXTSIZE]

&RETURN
/* #####
/* #####          ROUTINE PUT_USER_SPECS          #####
/* #####

/* RESTORES THE USER'S SPECIFICATIONS

&ROUTINE PutUserSpecs

    /* RESTORE THE USER'S NUMBER OF DECIMALS
    &FORMAT %UsersDecimals%

    /* RESTORE THE USER'S DISPLAY WINDOW SPECIFICATION
    DISPLAY %UsersAPdisplay%

    /* RESTORE THE USER'S SHADESET
    SHADESET %UsersShadeSet%

    /* RESTORE THE USER'S TEXT SPECIFICATIONS
    TEXTFONT %UserTextFont%
    TEXTQUALITY %UserTextQuality%
    TEXTSPACING %UserTextSpacing%
    TEXTSIZE %UserTextSize%

&RETURN

```

```

/* #####
/* #####          ROUTINE CHECK_INPUT_ERRORS          #####
/* #####

&ROUTINE CheckInputErrors

/* TODO: this is just one of the errors.  I should check for the rest

&IF %.MapsPerFrame% NE 1 AND ~
    %.MapsPerFrame% NE 2 AND ~
    %.MapsPerFrame% NE 3 AND ~
    %.MapsPerFrame% NE 4 AND ~
    %.MapsPerFrame% NE 6 &THEN
&DO
    &TYPE WARNING !!!!
    &TYPE You have chosen to display %.MapsPerFrame% grids per frame
    &TYPE Please, Choose to display 1, 2, 3, 4 or 6 grids per frame
    &TYPE
    &CALL bailout
&END

&RETURN

/* #####
/* #####          ROUTINE BAIL_OUT          #####
/* #####

&ROUTINE bailout

    &IF NOT [variable str] &THEN &S str Bailing out of %aml$file%
    &RETURN; &RETURN &WARNING %str%
    &TYPE

&RETURN

```