Chapter 2

Skeleton Tasks

By far the easiest way to write a new task is to find an old one that does something similar to what is desired and change it. With this thought in mind, we have written tasks whose sole purpose is to be changed into something useful. These tasks take care of most of the bookkeeping chores and make certain limited classes of operations quite simple. The source code for these tasks is heavily commented to aid the user in making the necessary modifications. The names and functions of these tasks are given in the following list.

- FUDGE This task modifies an existing uv data base and writes a new one.
- TAFFY This task modifies an existing image file and writes a new one.
- UVFIL This task creates, catalogs and fills a new uv data file.
- CANDY This task creates, catalogs and fills a new image file.
- PRPLn These tasks (PRPL1, PRPL2, PRPL3) are used to generate plots and are discussed in detail in the chapter on plotting.

Note: for many purposes task FETCH is adequate for reading an image into AIPS without modification. FETCH reads an image from a text file containing a description of the image. See the HELP file for FETCH for details.

Since these tasks contain most of the startup, shutdown, cataloging, etc. chores, they are a good place to start writing a new task. Many of the standard AIPS tasks are cloned from FUDGE or TAFFY. No one in the AIPS programming group has written a task from scratch in years. If the modified version of one of these tasks is to be of more than temporary use, the name of the task should be changed to avoid confusion. This chapter will describe in some detail the structure and use of the skeleton tasks.

2.1 Data Modification Tasks — FUDGE and TAFFY

There are two data modification tasks for the two types of data files, uv data (FUDGE) and images (TAFFY). The basic structure of these two tasks is very similar. The main routine in these tasks is very short and calls routines to do the basic functions:

1. Startup (FUDGIN in FUDGE, TAFIN in TAFFY)
    - initialize commons
    - get adverb values
    - restart AIPS (if DOWAIT is FALSE)
    - find input file in catalog
    - create and catalog output file
2. Process data (SENDUV in FUDGE, SENDMA in TAFFY)

3. write history (FUGHIS in FUDGE, TAFHIS called from OUTMA in TAFFY)
4. Shut down (DIE)
   - unmark catalog file statuses
   - restart AIPS if not done previously

Both FUDGE and TAFFY send one logical record (a visibility record in uv data or a row of an image) at a time to a user supplied subroutine. This subroutine can do some operation on the logical record and return the result. The result is then written to an output file. When all of the data has been processed, a final call is made to the user routine. In this call, the routine can record any entries to be made in the history file. In the history routine, the old history file is copied to the new file and some standard history entries are made. Then any user supplied entries are added. More detailed descriptions of FUDGE and TAFFY can be found in the following sections

2.1.1 FUDGE

FUDGE sends uv data records to a user supplied routine one at a time. The user routine performs some operation on the record and returns the record with a flag which says whether the result is to be kept or ignored. Many operations which require operating on several data records can be done by sorting the data with UVSRT so that records which are to be combined are adjacent in the data file. The structure of visibility records is described in detail in the chapter on disk I/O (Chapter 6).

If the size of the visibility record is unchanged, the only changes needed in FUDGE for most simple operations are in the user supplied routine DIDDLE. If the record size is changed, there must be changes made in FUDGIN so that the output file created has the correct size and catalog header information. SENDUV must also be modified so that it writes correct size records to the output file.

The source code for DIDDLE contains precursor comments explaining the use of the routine; these comments are reproduced below.

```
      SUBROUTINE DIDDLE (NUMVIS, U, V, W, T, IA1, IA2, VIS, RPARM,
     *   INCX, IRET)
C-----------------------------------------------------------------------
C   This is a skeleton version of subroutine DIDDLE which allows the
C   user to modify a UV data base.  Visibilities are sent one at a time
C   and when returned are written on the output file if so specified.
C
C   Up to 10 history entries can be written by using WRITE to
C   record up to 64 characters per entry into array HISCRD.  Format:
C          WRITE (HISCRD(entry #),format) history information
C   The history is written after the last call to DIDDLE.
C
C   Messages can be written to the monitor/logfile by encoding
C   the message (up to 80 char) into array MSGTXT in INCLUDE DMSG.INC
C   and then issuing a call:
C            CALL MSGWRT (priority #)
C
C   If IRET > 0, then the output file will be destroyed iff
C   it was created in the current execution.
C
C   If the size of the vis record is to be changed, appropriate
C   modifications should be made to CATBLK in FUDGIN before the call
C   to UVCREA and LRECO in SENDUV should reflect the correct size of
C   the output record.
C
C   See the precursor comments for UVPGET for a description
```

```
C   of the contents of COMMON /UVHDR/ which allows easy access to
C   much of the information from the catalog header (CATBLK) and
C   which describes the order in which the data is given.
C
C   After all data has been processed a final call will be made to
C   DIDDLE with NUMVIS = -1.  This is to allow for the completion of
C   pending operations, i.e. preparation of HIstory cards.
C
C   LUN's 16 and 17 are open and not available to DIDDLE.
C
C   The current contents of CATBLK will be written back to the
C   catalog after the last call to DIDDLE.
C
C   Inputs:
C      NUMVIS  I    Visibility number, -1 => final call, no data
C                   passed but allows any operations to be completed.
C      U       R    U in wavelengths
C      V       R    V in wavelengths
C      W       R    W in wavelengths
C      T       R    Time in days since 0 IAT on the first day for
C                   which there is data, the julian day corresponding
C                   to this day can be obtained in D   form by:
C                   CALL JULDAY (CATH(KHDOB),XDAY) where XDAY will
C                   be the Julian day number.
C      IA1     I    First antenna number
C      IA2     I    Second antenna number
C      RPARM   R(*) Random parameter array which includes U,V,W etc
C                   but also any other random parameters.
C      VIS     R(INCX,*)  Visibilities in order real, imaginary, weight
C                   (Jy, Jy, unitless).  Weight <= 0 => flagged.
C                   NOTE: INCX may be any value .GE. 2
C   Inputs from COMMON:
C      NAME2      C*12    Name of the aux. file
C      CLAS2      C*6     Class of the aux. file.
C      SEQ2       I       Sequence number of the aux. file.
C      DISK2      I       Volumn number of the aux. file.
C      APARM(10)  R(10)   User array.
C      BPARM(10)  R(10)   User array.
C      BOX(4,10)  R(4,10) User array.
C      RA         D       Right ascension (1950) of phase center. (deg)
C      DEC        D       Declination (1950) of phase center. (deg)
C      FREQ       D       Frequency of observation (Hz)
C      NRPARM     I       # random parameters.
C      NCOR       I       # correlators
C      CATBLK     I(256)  Catalog header record.
C
C   Output:
C      U          R    U in wavelengths
C      V          R    V in wavelengths
C      W          R    W in wavelengths
C      T          R    Time in same units as input.
C      RPARM      R    Modified random parameter array.  N.B.  U,V,W,
C                      time, baseline should not be modified in RPARM
C      VIS        R    Visibilities
```

```
C      IRET       I    Return code  -1 => don't write
C                                    0 => OK
C                                   >0 => error, terminate.
C
C   Output in COMMON:
C      NUMHIS    I         # history entries (max. 10)
C      HISCRD    C(NUMHIS) History records
C      CATBLK    I         Catalog header block
C-----------------------------------------------------------------------
```

There are a number of adverbs already included in FUDGE to pass user information to the user routine; these are specifications for a second input file and the arrays CPARM, DPARM and BOX. More or different adverbs are readily added.

FUDGE will automatically compress the output file if the number of visibility records in the file is reduced. The source code for FUDGE can be found in the standard program source area; this is usually assigned the logical name "APLPGM:" whose value is AIPS_VERSION:[APL.PGM] on VMS systems.

2.1.2 TAFFY

TAFFY reads a selected subset (or all) of an image, sends the image one row at a time to a user supplied routine (DIDDLE) which operates on the row. The user routine sends back the result which may be of arbitrary length; in particular the input row may be reduced to a single value. The values sent back from the user supplied routine are written into the new cataloged file. DIDDLE can defer returning the next row; this allows the use of scrolling buffer. TAFFY can handle multi-dimensional, blanked images. The task TRANS may be used before a TAFFY clone to transpose which ever axis is necessary to the first axis. The returned value of a row may be deferred for those cases when a scrolling buffer of the input is needed.

If the size or format of the output file is to be different from the input file, or if it is necessary to check that the proper axis occurs first in the data array, or if there are several possible operations to be specified by the adverb OPCODE, then the routine NEWHED needs to be modified. The main purpose of NEWHED is to form the catalog header record for the output file. For many purposes the only modifications needed to NEWHED are to modify the values in DATA statements from the default values supplied. The beginning portion of NEWHED is reproduced below.

```
      SUBROUTINE NEWHED (IRET)
C-----------------------------------------------------------------------
C   NEWHED is a routine in which the user performs several operations
C   associated with beginning the task.  For many purposes simply
C   changing some of the values in the DATA statments will be all that
C   is necessary.  The following functions are/can be performed
C   in NEWHED:
C       1) Modifying the catalog header block to represent the
C   output file.  The MINIMUM modifications required here are those
C   required to define the size of the output file; ie.
C      CATBLK(KIDIM)   = the number of axes,
C      CATBLK(KINAX+i) = the dimension of each axis, and
C   Other changes can be made either here or in DIDDLE; the
C   catalog block will be updated when the history file is
C   written.
C       2) Checking the input image and/or input parameters.
C   For example, if a given first axis type such as
C   Frequency/Velocity is required this should be checked.  The
C   routine currently does this and all that is required to
C   implement this is to modify the DATA statments.
C   A returned value of IRET .NE. 0 will cause the task to terminate.
```

```
C   A message to the user via MSGWRT about the reason for the
C   termination would be friendly.  This can be done by encoding
C   the message into MSGTXT, setting IRET to a non-zero value
C   and issuing a GO TO 990.
C       3) Default values of some of the input parameters
C   (OUTNAME, OUTCLASS, OUTSEQ, OUTDISK, TRC and BLC defaults are
C   set elsewhere).  As currently set the default OPCODE is the
C   first value in the array CODES which is set in a data statment.
C
C    Input in common:
C       CATBLK    I(256)  Output catalog header, also CATR, CATD
C       CATOLD    I(256)  Input catalog header, also OLD4, OLD8
C    Output:
C       CATBLK    I(256)  Modified output catalog header.
C       IRET      I       Return error code, 0=>OK, otherwise abort.
C-----------------------------------------------------------------------
      INTEGER   IRET
C
      CHARACTER ATYPES(10)*8, FCHARS(3)*4, BLANK*8, CODES(10)*4,
     *   UNITS(10)*8, CTEMP*8
      HOLLERITH OLD4(256)
      DOUBLE PRECISION    OLD8(128)
      INTEGER   NCODE, NTYPES, IOFF, IERR, INDXI, INC, INDEX,
     *   NCHTYP(10), LIMIT, I, FIRSTI, FIRSTO
      LOGICAL   LDROP1
      INCLUDE 'INCS:DDCH.INC'
      INCLUDE 'INCS:DMSG.INC'
      INCLUDE 'INCS:DHDR.INC'
      INCLUDE 'TAFFY.INC'
      INCLUDE 'INCS:DCAT.INC'
      EQUIVALENCE (CATOLD, OLD4, OLD8)
      DATA FCHARS /'FREQ','VELO','FELO'/
      DATA BLANK /'        '/
C                                       User definable values
C                                       # and value of OPCODEs
      DATA NCODE /0/
      DATA CODES /10*'    '/
C                                       Output units for each OPCODE.
      DATA UNITS /'UNDEFINE',9*'    '/
C                                       Allowed number of axis types
C                                       and types.
      DATA NTYPES /0/
      DATA ATYPES /10*'        '/
      DATA NCHTYP /10*4/
C                                       If LDROP1 is .TRUE. then the
C                                       first axis will be dropped,
C                                       (ie, one value results from
C                                       the operation on each row.)
      DATA LDROP1 /.FALSE./
```

The data modification routine in TAFFY is DIDDLE which contains numerous precursor comments describing its use; these precursor comments follow.

```
      SUBROUTINE DIDDLE (IPOS, DATA, RESULT, IRET)
```

```
C-----------------------------------------------------------------------
C   This is a skeleton version of subroutine DIDDLE which allows
C   operations on an image one row at a time (1st dimension).
C   Input and output data may be blanked. The calling routine keeps
C   track of max., min. and the occurence of blanking.  If DROP1 is
C   .TRUE., the calling routine expects 1 value returned per call;
C   otherwise, CATBLK(KINAX) values per call are expected returned.
C   NOTE: blanked values are denoted by the value of the common variable
C   FBLANK.
C       DIDDLE may accumulate a scrolling buffer by returning a negative
C   value of IRET.  This tells the calling routine to defer writing the
C   next row.  If rows are deferred then an equal number of calls to
C   DIDDLE will be made with no input data; this allows reading out any
C   rows left in DIDDLEs internal buffers.  Such a "no input call" is
C   indicated by a value of IPOS(1) of -1.  The writing of the returned
C   values of these "no input calls" may NOT be deferred.
C       Up to 10 history entries can be written  to
C   record up to 64 characters per entry into array HISCRD. Ex:
C         WRITE (HISCRD(entry #), format) list
C   TRC, BLC and OPCODE are already taken care of.
C   The history is written after the last call to DIDDLE.
C       Messages can be written to the monitor/logfile by encoding
C   the message (up to 80 char) into array MSGTXT in COMMON /MSGCOM/
C   and then issuing a call:
C        CALL MSGWRT (priority #)
C
C       If IRET .GT. 0 then the output file will be destroyed.
C
C       After all data have been processed a final call will be made to
C   DIDDLE with IPOS(1)=-2.  This is to allow for the completion of
C   pending operations, i.e. preparation of HIstory cards.
C
C       AIPS LUN's 16-18 are open and not available to DIDDLE.
C
C       The current contents of CATBLK will be written back to the
C   catalog after the last call to DIDDLE.
C
C   Inputs:
C      IPOS   I(7)    BLC (input image) of first value in DATA
C                     IPOS(1) = -1 => no input data this call.
C                     IPOS(2) = -2 => last call (no input data).
C      DATA   R(*)    Input row, magic value blanked.
C   Values from commons:
C      ICODE   I      Opcode number from list in NEWHED.
C      FBLANK  R      Value of blanked pixel.
C      CPARM   R(10)  Input adverb array.
C      DPARM   R(10)  Input adverb array.
C      CATBLK  I      Output catalog header (also CATR, CATD)
C      CATOLD  I      Input catalog header (also OLD4, OLD8)
C      DROP1   L      True if one output value per call.
C   Output:
C      RESULT  R(*)   Output row.
C      IRET    I      Return code   0 => OK
C                                  >0 => error, terminate.
```

```
C   Output in COMMON:
C     NUMHIS  I          # history entries (max. 10)
C     HISCRD  C(NUMHIS)  History records
C     CATBLK  I          Catalog header block
C-----------------------------------------------------------------------
```

In addition to the adverb OPCODE to specify the desired operation and the adverbs BLC and TRC to specify the window in the input map, there are several user defined adverbs sent to TAFFY. These are the arrays CPARM and DPARM; more and/or other adverbs can be added.

More details about TAFFY can be found in the comments in the source version of the program. The source code for TAFFY can be found in the standard program source area; this is usually assigned the logical name "APLPGM:" whose value is AIPS_VERSION:[APL.PGM] on VMS systems.

2.2 Data Entry Tasks (UVFIL and CANDY)

There is a pair of skeleton tasks for entering data into AIPS, UVFIL for uv data and CANDY for images. These tasks are used to enter either observational or model data into the AIPS system. CANDY especially has been used a number of times and usually takes a couple of hours to produce a working program. (Use of task FETCH is useful in many cases for entering an image into AIPS).

These tasks each have two subroutines which may need to be supplied or modified. The first routine is the one to create the new header record and, for UVFIL, to enter information about the antennas. Most of the modifications required are changes to DATA statements from the supplied default values. The beginning portion of these routines will be given with the detailed descriptions of UVFIL and CANDY. Details about the catalog header record are given in the chapter on catalogs.

The second routine, to be supplied by the user, generates the data to be written to the output file. This may be done by reading an external disk or tape file or by any other means.

The basic structure of UVFIL and CANDY are very similar. The main routine in these tasks is very short and calls routines to do the basic functions:

1. Startup (UVFILN in UVFIL, CANIN in CANDY)
    - initialize commons
    - get adverb values
    - restart AIPS (If DOWAIT is FALSE)
2. Create new catalog header record (NEWHED)
    - create and catalog output file
    - Enter antenna information (In UVFIL only)
3. Read/generate data (FIDDLE in UVFIL, MAKMAP in CANDY)
4. Write history (and antenna file) (FILHIS in UVFIL, CANHIS in CANDY)
5. Shut down (DIE)
    - Unmark catalog file statuses
    - Restart AIPS if not done previously

2.2.1 UVFIL

UVFIL creates, catalogs and fills an AIPS uv data file. It can be used either to translate uv data from another format or generate model data.

UVFIL comes with specific example code reading a file. The first routine, NEWHED, which the user may need to modify is used to enter information required to create the catalog header block and to enter information about the antennas. The beginning portion of this routine follows:

```
      SUBROUTINE NEWHED (IRET)
C-----------------------------------------------------------------------
C   NEWHED is a routine in which the catalog header is constructed.
C   Necessary values can be read in in the areas markes "USER CODE
C   GOES HERE".
C
C   NOTE: the AIPS convention for the coordinate reference value
C   for the STOKES axis is that 1,2,3,4 represent I, Q, U, V
C   stokes' parameters and -1,-2,-3,-4 represent RR, LL, RL and
C   LR correlator values.  Currently set for R and L polarization
C   ie Ref. value = -1 and increment = -1.
C
C   The MINIMUM information required here is that
C   required to define the size of the output file; ie.
C      CATBLK(KIGCN) = Number of visibility records
C      CATBLK(KIPCN) = Number of random parameters.
C      CATBLK(KIDIM) = Number of axes,
C      CATBLK(KINAX+i) = the dimension of each axis.
C   Other changes can be made either here or in FIDDLE; the
C   catalog block will be updated when the history file is
C   written.
C      The antenna information can also be entered in this
C   routine.  It is possible to put much more information in the
C   ANtenna file.
C
C   Input in common:
C      CATBLK(256)    I     Output catalog header, also CATR, CATH, CATD
C                           The OUTNAME, OUTCLASS, OUTSEQ are entered
C                           elsewhere.
C   Output in common:
C      CATBLK(256)    I     Modified output catalog header.
C      IRET           I     Return error code, 0=>OK, otherwise abort.
C    Also the antenna information can be filled into a common.
C-----------------------------------------------------------------------
      INTEGER   IRET
C
      CHARACTER RTYPES(7)*8, TYPES(7)*8, UNITS*8, TELE*8, OBSR*8,
     *   INSTR*8, OBSDAT*8, LINE*80
      INTEGER   I, NAXIS, NRAN, NCHAN, NPOLN, NDIM(7), INDEX, XCOUNT,
     *   LUN, FIND
      LOGICAL   APPEND
      REAL      CRINC(7), CRPIX(7), EPOCH, BANDW
      DOUBLE PRECISION CRVAL(7)
      INCLUDE 'UVFIL.INC'
      INCLUDE 'INCS:DCAT.INC'
      INCLUDE 'INCS:DMSG.INC'
      INCLUDE 'INCS:DHDR.INC'
```

```
      INCLUDE 'INCS:DUVH.INC'
C                                       User definable values
C                                       Random parameters.
C                                         No. random parameters.
      DATA NRAN /5/
C                                         Rand. parm. names.
      DATA RTYPES /'UU-L-SIN','VV-L-SIN','WW-L-SIN',
     *   'TIME1   ','BASELINE',2*'        '/
C                                       Uniform axes.
C                                         No. axes.
      DATA NAXIS /5/
C                                         Axes names.
      DATA TYPES /'COMPLEX ','STOKES  ','FREQ    ',
     *   'RA      ','DEC     ',2*'        '/
C                                         Axis dimensions
      DATA NDIM /3,1,1,1,1,0,0/
C                                         Reference values
      DATA CRVAL /1.0D0, -1.0D0, 5*0.0D0/
C                                         Reference pixel.
      DATA CRPIX /7*1.0/
C                                         Coordinate increment.
      DATA CRINC /1.0, -1.0, 0.0, 0.0, 0.0, 2*0.0/
C                                       Epoch of position.
      DATA EPOCH /1950.0/
C                                       Units
      DATA UNITS /'JY      '/
C-----------------------------------------------------------------------
```

The user supplied routine FIDDLE returns visibility records which are written into the cataloged output file. The precursor comments describing the use of FIDDLE follow.

```
      SUBROUTINE FIDDLE (NUMVIS, U, V, W, T, IA1, IA2, VIS, RPARM, IRET)
C-----------------------------------------------------------------------
C  This is a skeleton version of subroutine FIDDLE which allows the
C  user to create a UV data base.  Visibilities are returned one at
C  a time and are written on the output file.
C
C       Up to 10 history entries can be written by using WRITE to
C  record up to 64 characters per entry into array HISCRD. Ex:
C     WRITE (HISCRD(entry #),format #) list
C  The history is written after the last call to FIDDLE.
C
C       Messages can be written to the monitor/logfile by writing
C  the message (up to 80 char) into array MSGTXT in INCLUDE DMSG.INC
C  and then issuing a call:
C     CALL MSGWRT (priority #)
C
C       If IRET .GT. 0 then the output file will be destroyed.
C   A value of IRET .lt. 0 indicates the end of the data.
C
C       See the precursor comments for UVPGET for a description
C  of the contents of COMMON /UVHDR/ which allows easy access to
C  much of the information from the catalog header (CATBLK) and
C  which describes the order in which the data is being written.
```

```
C
C      After all data has been processed a final call will be made to
C  FIDDLE with NUMVIS = -1.  This is to allow for the completion of
C  pending operations, i.e. preparation of HIstory cards.
C
C    AIPS I/O  LUN 16 is open and not available to FIDDLE.
C  FORTRAN unit numbers greater than 50 will probably not get the
C  AIPS routines confused.  (Any unit numbers other than 1, 5, 6 and 12
C  will probably also work.)
C
C      The current contents of CATBLK will be written back to the
C  catalog after the last call to FIDDLE.
C
C  Inputs:
C    NUMVIS     I    Visibility number, -1 => final call, no data
C                    passed but allows any operations to be completed.
C
C  Inputs from COMMON:
C    IN2FIL  C*48  Name of the aux. file
C    APARM   R(10) User array.
C    BPARM   R(10) User array.
C    RA         D  Right ascension (1950) of phase center. (deg)
C    DEC        D  Declination (1950) of phase center. (deg)
C    FREQ       D  Frequency of observation (Hz)
C    NRPARM     I  # random parameters.
C    NCOR       I  # correlators
C    CATBLK(256)I  Catalog header record. See Going AIPS for details.
C
C  Output:
C    U       R       U in wavelengths at the reference frequency.
C    V       R       V in wavelengths
C    W       R       W in wavelengths
C    T       R       Time in days since the midnight at the start of
C                    the reference date.
C    IA1     I       Antenna number of the first antenna.
C    IA2     I       Antenna number of the second antenna.
C                    NOTE: IA2 MUST be greater that IA1
C    RPARM   R       Modified random parameter array. NB U,V,W,
C                    time and baseline should not be modified in RPARM
C    VIS     R(3,*)  Visibilities.  The first dimension is the COMPLEX
C                    axis in the order Real part, Imaginary part,
C                    weight. The order of the following visibilities is
C                    defined by variables in COMMOM /UVHDR/ (originally
C                    specified in NEWHED).  The order number for Stokes
C                    parameters is JLOCS and the order number for
C                    frequency is given by JLOCF.  The lower order
C                    number  increases faster in the array.
C                    See precursor comments in UVPGET for more details.
C    IRET    I       Return code  -1 => End of data.
C                            0 => OK
C                           >0 => error, terminate.
C
C  Output in COMMON:
C    NUMHIS   I          # history entries (max. 10)
```

```
C     HISCRD   C(NUMHIS)  History records
C     CATBLK   I          Catalog header block
C-----------------------------------------------------------------------
```

The user defined array adverbs APARM and BPARM are sent to UVFIL; more and/or other adverbs can easily be added. The source code for UVFIL can be found in the non-standard program source area; this is usually assigned the logical name "APGNOT:" whose value is AIPS_VERSION:[APL.PGM.NOTST] on VMS machines.

2.2.2 CANDY

CANDY is similar to TAFFY except there is no AIPS input data file. This is a good routine to use to generate an AIPS image from either a model or an external data file. CANDY has example code (mostly commented out) in the text which gives an example of reading a formatted disk file. (Note this function is also done in a general way in routine FETCH).

The routine in CANDY in which the values necessary for the catalog header must be entered is named NEWHED. The beginning, heavily commented, portion of NEWHED follows.

```
      SUBROUTINE NEWHED (IRET)
C-----------------------------------------------------------------------
C   NEWHED is a routine in which the user performs several operations
C   associated with beginning the task.  For many purposes simply
C   changing some of the values in the DATA statments will be all that
C   is necessary.  The following functions are/can be preformed
C   in NEWHED:
C       1) Creating the catalog header block to represent the
C   output file.  The MINIMUM information required here is that
C   required to define the size of the output file; ie.
C      CATBLK(KIDIM)= the number of axes,
C      CATBLK(KINAX+i) = the dimension of each axis.
C   Other changes can be made either here or in MAKMAP; the
C   catalog block will be updated when the history file is
C   written.
C       2) Setting default values of some of the input parameters
C   As currently set the default OPCODE is the first value in the
C   array CODES which is set in a data statment.
C
C    Input:
C     CATBLK    I(256)  Output catalog header, also CATR, CATD
C                       The OUTNAME, OUTCLASS, OUTSEQ are entered
C                       elsewhere.
C    Output:
C     CATBLK    I(256)  Modified output catalog header.
C     IRET      I       Return error code, 0=>OK, otherwise abort.
C-----------------------------------------------------------------------
      INTEGER   IRET
C
      CHARACTER FCHARS(3)*4, BLANK*8, CODES(10)*4, UNITS(10)*8,
     *   ATYPES(7)*8, LINE*80
      INTEGER   I, NAXIS, IROUND, NCODE, IERR, NX, NY, INDEX
      INCLUDE 'CANDY.INC'
      INCLUDE 'INCS:DCAT.INC'
      INCLUDE 'INCS:DDCH.INC'
      INCLUDE 'INCS:DMSG.INC'
```

```
      INCLUDE 'INCS:DHDR.INC'
      DATA FCHARS /'FREQ','VELO','FELO'/
      DATA  BLANK /'        '/
C                                       User definable values
C                                       # and value of OPCODEs
      DATA NCODE /0/
      DATA CODES /10*'    '/
C                                       Output units for each OPCODE.
      DATA UNITS /'UNDEFINE',9*'        '/
C                                       Number of axes and types.
C                                       (Set for two axes = Ra, Dec.)
      DATA NAXIS  /2/
      DATA ATYPES /'RA---SIN', 'DEC--SIN',
     *   'STOKES  ', 'FREQ    ', 3*'        '/
C-----------------------------------------------------------------------
```

The user supplied routine that reads or generates the image is MAKMAP. This routine returns the image one row at a time. The precursor comments describing the use of this routine follow.

```
      SUBROUTINE MAKMAP (IPOS, RESULT, IRET)
C-----------------------------------------------------------------------
C  This is a skeleton version of subroutine MAKMAP which allows
C  the user to create an image, one row at a time.
C  Output values may  be blanked.
C  The calling routine keeps of max., min. and to occurence of blanking.
C  CATBLK(KINAX) values per call are expected returned.
C  NOTE: blanked values are denoted by the value of the common variable
C  FBLANK
C
C      Up to 10 history entries can be written by using WRITE to
C  record up to 64 characters per entry into array HISCRD. Ex:
C     WRITE (HISCRD(entry #),format #,) list
C  TRC, BLC and OPCODE are already taken care of.
C  The history is written after the last call to MAKMAP.
C
C      Messages can be written to the monitor/logfile by writing
C  the message (up to 80 char) into array MSGTXT in COMMON /MSGCOM/
C  and then issuing a call:
C     CALL MSGWRT (priority #)
C
C      If IRET .GT. 0 then the output file will be destroyed.
C
C      After all data has been processed a final call will be made to
C  MAKMAP with IPOS(1)=-1.  This is to allow for the completion of
C  pending operations, i.e. preparation of HIstory cards.
C
C      LUN's 16-18 are open and not available to MAKMAP.
C
C      The current contents of CATBLK will be written back to the
C  catalog after the last call to MAKMAP.
C
C  Inputs:
C   IPOS   I(7)    BLC (input image) of first value in DATA
C  Values from commons:
```

```
C   ICODE   I      Opcode number from list in NEWHED.
C   FBLANK  R      Value of blanked pixel.
C   CPARM   R(10)  Input adverb array.
C   DPARM   R(10)  Input adverb array.
C   CATBLK  I(256) Output catalog header (also CATR, CATD)
C  Output:
C   RESULT   R(*)  Output row.
C   IRET     I     Return code   0 => OK
C                               >0 => error, terminate.
C  Output in COMMON:
C  NUMHIS  I          # history entries (max. 10)
C  HISCRD  C(NUMHIS)  History records
C  CATBLK  I          Catalog header block
C-----------------------------------------------------------------------
```

Pixel blanking is supported through magic value blanking, i.e., the value of FBLANK is recognized to mean no value is associated with the pixel. The source code for CANDY is fairly heavily commented and can be found in the non-standard program source area; this is usually assigned the logical name "APGNOT:" whose value is AIPS_VERSION:[APL.PGM.NOTST] on VMS systems.

2.3 Modifying a Skeleton Task

To make a modified version of one of the skeleton tasks, first copy the source code and the help file to the area in which you intend to work on the task. Then rename the task to avoid confusion (only five characters are allowed in an AIPS task name). In addition to changing the name of the files, it is crucial to change the name of the task entered in a DATA statement in the main program. You should also change the task name referenced in the help file. (If there is a chance that your new task will become part of the standard AIPS package, and we welcome contributions, rename the names of the subroutines as well.)

The next step is to modify the source code to taste. If the adverbs which the task uses are changed, the help file should also be changed to reflect this. If the task is to be of more than temporary use, then it is friendly to put sufficient documentation into the help file to assist other users in understanding the use of the input adverbs; besides, you will also forget just what it is that BPARM(3) does. Once the source code is modified, see Appendix A for details about compiling, linking and debugging a task

THIS PAGE DELIBERATELY BLANK