Appendix D: Cray Job Decks

D.1 Purpose

The job decks (shell scripts) are used to run the MM5 modeling system programs. Job decks for NCAR Crays (paiute, ouray, or chipeta) are written in C-shell, and can be used in either batch or interactive mode. Users with syntax questions for C-shell constructs should refer to man page for "csh". New Cray users should browse through NCAR/Scientific Computing Division's (SCD) UNICOS users' guide (URL: *http://www.scd.ucar.edu/docs/UNICOS/unicos_guide.toc.html*) to get familiar with the Cray environment. Questions regarding Cray usage should be directed to SCD consultant at *consult1@ucar.edu*. For information on obtaining an computer account at NCAR, please visit SCD's Web page: *http://www.scd.ucar.edu/css/resources/requesthome.html.*

The modeling system job decks are structured similarly to each other. The discussion in this section is focused on describing the general structure of the job deck. Shell variables that are common throughout the MM5 modeling system job decks are discussed. The examples are taken from the TERRAIN, DATAGRID, RAWINS, INTERP, GRAPH, and MM5 job decks located in the

~mesouser/MM5V2

directory seen on ouray/paiute/chipeta or /mesouser/MM5V2 directory on NCAR's anonymous ftp (ftp.ucar.edu).

D.2 Prerequisite

There are several things a user needs to prepare before starting running jobs on NCAR's Crays.

- Make sure your .cshrc file is up-to-date and contains the following:

```
# Initialize modules
     if (-f /opt/modules/modules/init/csh ) then
          source /opt/modules/modules/init/csh
          module load modules PrgEnv
          module load modules nqe
     endif
```

and

```
     setenv NCARG_ROOT /usr/local
```

This enables a user to use Cray utilities (such as the compiler) and NCAR Graphics. The environment variable **NCARG_ROOT** is required on a workstation where NCAR Graphics is available.

- Make sure you have an up-to-date .rhosts file on both your local machine and Crays. A typical .rhosts file looks like this:

```
paiute.ucar.edu username
ouray.ucar.edu username
```

- Make sure that you browse through the ***~mesouser*** directories on the Crays, or ***mesouser/*** directory on anonymous ftp. All job decks, program tar files, data catalogs, and utility programs reside in the directory.

D.3 Functions of Job Decks

The general job deck construct and functions are the following:

- batch job commands that are used for Crays to request CPU time and memory
- input/output pathnames to retrieve and archive data files
- job switches and output options
- where to obtain Fortran program tar files
- parameter statements used in Fortran programs to define domain and data dimensions
- FORTRAN namelist used during program execution to select runtime options
- a section that does not normally require user modification

All Cray job decks are structured similarly. Each deck begins with a number of QSUB commands required for batch job execution on a Cray. It is followed by shell variables needed to set up data input and output pathnames, job switches, and your source code location, etc. The next section of a job deck contains PARAMETER statements required for Fortran programs, and namelist used during execution. The remaining job deck, which handles the data input, generation of the executable from the FORTRAN source code, and data archival, does not typically need user modification. A user needs only to modify this section to adjust the default setting for running the modeling system on other site.

If a user plans to work on NCAR's Crays, all he/she needs to acquire are the job decks from the ***~mesouser/MM5V2*** directory. These decks will enable a user to run in batch as well as in interactive mode on the Cray. When using these decks interactively, cd to $TMPDIR (this directory will disappear when one logs off), and copy the tar files to the directory before running.

D.4 What to Modify in a Job Deck?

D.4.1 Batch Job Queueing Commands

Each of the Cray job decks has several lines of header commands for the UNICOS network queueing system. Inclusion of these commands allows the decks to be submitted into the batch queue locally on the Cray with "qsub" command (e.g. *qsub my.job)*, or remotely from a user's front-end machine through IRJE or MIGS (see usage on *http://www.scd.ucar.edu/docs/catalog/*). The QSUB commands inside the job decks are preceded by the "#" character; when these jobs are run interactively, the QSUB commands are interpreted as comments.

```
# QSUB -r DATAGRID              # request name
# QSUB -reg                     # job queue class
# QSUB -eo                      # stdout and stderr together
# QSUB -lM 5Mw                  # maximum memory
# QSUB -lT 300                  # time limit per request
# QSUB                          # no more qsub commands
#
ja
set echo
#
```

```
# Define the batch output name
#
batchname datagrid.out
#
#           *******************************************
#           ********    datagrid batch C shell   *******
#           ********       V2 System             *******
#           ********    for both cf77 and f90    *******
#           *******************************************
#
#   This version of the DATAGRID job deck can process the model first guess
#   fields for all domains in one DATAGRID submittal
#
......
#
if ( $ENVIRONMENT == BATCH ) then
   cd $TMPDIR
endif
```

#QSUB "option value"

This syntax is a shell command that specifies the UNICOS Network Queuing System (NQS) options for batch jobs. If the script is executed interactively, the QSUB will be ignored by the shell. The NQS 'option values' listed in the shell script that require user's modification are described below

#QSUB -q queue:

This option sets the queue. The available NCAR/Cray job queues are listed in NCAR SCD documentation (the URL is *http://www.scd.ucar.edu/docs/UNICOS/04.nqs.html*).

#QSUB -lM nMw:

This sets the maximum memory used by the job. The value n has the unit of mega words. For non-dynamical-memory-allocated programs, the memory required by a job can be estimated by using the '*size*' command: *size program.exe*, for example.

#QSUB -lT n:

This is the maximum CPU time, in seconds, required by the entire script (including data transfer to and from MSS, no grace period).

If the program begins execution and subsequently fails with either of the two following error messages,

```
not enough space
cpu limit exceeded
```

the user should resubmit the job deck after augmenting the memory allocation to increase space (#**QSUB -lM)** and/or the CPU time limit **(#QSUB -lT).**

ja:

This UNICOS command starts the UNICOS job accounting information.

set echo:

Echoes commands executed in the script. Useful for checking job status when it fails.

cd $TMPDIR:

When this command is executed in a batch job deck, a temporary directory is created when the job execution begins. The directory has a name nqs.+++++XXXX, where XXXX are an assigned number/letter combination. No files are scrubbed from $TMPDIR until all of the initiated processes have completed, then all files are removed and the directory is deleted at the end of the job. (You can also use this command to create a temporary working directory when you log in. In this case, the directory will bear a name jtmp.XXXXXXa, where XXXXXX is a 6-digit integer. This directory will disappear when you log off.)

D.4.2 Shell Variables

After the QSUB commands, the next user modification is the shell variables that are used to set up NCAR Cray's Mass Storage System (MSS) pathnames, remote machine name for copying files (rcp), and job switches. The job decks have all of the optional settings available, with the inactive lines commented (with a # at the first column).

- MSS related set up

On NCAR's Cray, the input and output files generated by the MM5 modeling system are stored on MSS. Users need to set up the MSS pathname for the input and output files so the script can retrieve the input files prior to the execution of the program, and store the output files after the program is completed. For complete information on how to use the MSS facility at NCAR, please review SCD's online documentation: *http://www.scd.ucar.edu/docs/catalog/cat.mss.html*. Usage of MSS is charged against your GAUs allocation. If you are running the job decks on other Cray which doesn't have MSS, the MSS related shell variables can be defined as a pathname to a data disk. Example:

```
set ExpName = TEST/EXP1       # MSS pathname
set RetPd = 365               #  Retention period in days for MSWRITEs
#
#       MSS filenames for terrain data sets
#
set InTerr     =    ( ${ExpName}/TERRAIN_DOMAIN1 )
#
set InAnly = /DSS/xxxxxx      #  MSS Filename(s) for NMC analyses
set InSnow           # MSS Filename for additional snow-cover file (optional)
```

ExpName:

This is used to set up the MSS input and output pathname, and is recommended to use something that describes the experiment. A user should use the same ExpName for every job decks in the MM5 modeling system for a single experiment. If ExpName doesn't start with a /, then NCAR Crays' login name as MSS root name (upper case) is assumed.

RetPd:

It is used to set the retention period in days for the MSS files. This is one of the options for the local NCAR mswrite command. The upper limit for retention period is 32767 days.

InX, OutX:

One of the conventions used to denote input files from output files is the name given to the shell variable holding the MSS path locations. Shell variables starting with the string "In" are files coming from the MSS; they are inputs to a program. Shell variables starting with the string "Out" are to be achived to the MSS; they are outputs from a program. The same MSS file can be an output file from one of the job decks, and then be an input file in a subsequent program. In a normal submittal, the **InX** and **OutX** variables do not need to be changed. However, if a user needs to input a MSS file under other user's login name, these variables need to be modified. (Note that **InX** and **OutX** only provide the MSS pathname, the file names are hard-wired in the section of the shell scripts that does not typically require user modification. For example, an output file from program TERRAIN would be TEST/EXP1/TERRAIN_DOMAIN1.)

- RCP set up

The rcp command copies files between machines. In the deck, it is mostly used for situations such that you have a modified source code tar file located on your local machine, or getting some output files to your local machine (e.g. the plot file from TERRAIN, or auto bogus output from RAWINS). A shell variable **Host** is defined to set up the path for rcp:

```
set Host     = username@host.domain:directory
```

Host and domain:

They are used to remotely copy files to and from NCAR's Cray to user's local machine. The shell variable Host should have a form of **username@host.domain:directory**, where **username** is user's login name on the user's local machine, **host** and **domain** are local machine's name and address, and **directory** is the directory to copy a file to or from. If the user's local login name is the same as the login name on NCAR's Cray, username may be eliminated.

In order for the remote shell facility to work on another machine without logging on, such as rcp, you must have a file called *.rhosts* in the login directory on each remote machine that you want to copy files to or from. The *.rhosts* file should contain a list of machine and username pairs:

```
host1.domain1 username1
host2.domain2 username2
```

For example, if mesouser wants to have remote access between NCAR's ouray and local machine mmm1 and mmm2, the *.rhosts* file on ouray should have the following lines

```
mmm1.mmm.ucar.edu mesouser
mmm2.mmm.ucar.edu mesouser
```

and the *.rhosts* file on machines mmm1 and mmm2 should have the lines

```
ouray.ucar.edu mesouser
ouray mesouser
```

Note that in this example there are two entry lines for ouray. The user needs to test out on their local machine to see whether only one is needed. The user also needs to protect their *.rhosts* files to avoid security problem by using the following command

```
chmod 600 .rhosts
```

If mesouser wants to rcp a file back to machine mmm1 under directory /usr/tmp/mesouser, the Host variable should be set to

```
    set Host = mesouser@mmm1.mmm.ucar.edu:/usr/tmp/mesouser
```

- Job Switches

Since the MM5 modeling system is designed for multiple applications, there are many options on how a job may be run. These options include different sources for input meteorological data, ways to do objective analysis, running the model hydrostatically or nonhydrostatically, and whether the job is an initial or restart run, etc.. A user is required to go through the shell variables and make selection of job switches. The following is an example taken from the *datagrid.deck*, and the selection is with regarding to the type of global analysis to be used to create the first guess fields:

```
#
#       type of datagrid job
#
# set Analysis = NMC          # NMC first guess
# set Analysis = ECMWF         # Old ECMWF format first guess
set Analysis    = TOGA        # ECMWF TOGA format first guess
# set Analysis = UNIDATA       # Unidata first guess (NMC MRF)
#
```

This table lists the shell variables that need to be defined by users for each program of the MM5 modeling system:

Program Name	*Shell Variables*
TERRAIN	TerPlt
DATAGRID	Analysis, SST
RAWINS	Submit, INOBS, SFCsw, BOGUSsw
INTERP	HYDROsw, ForBsw, NESTsw
MM5	compile, execute, STARTsw, FDDAsw, HYDROsw

These shell variables will be discussed in detail in the other chapters of this document.

- IEEE format data output option

The IEEE data are standard data format to use on a workstation. An option to output data in IEEE format from the MM5 modeling system programs TERRAIN, DATAGRID, RAWINS, and INTERP is provided in the shell by setting

```
OutIEEE = yes
```

It makes use of a shell script *ieee.csh* copied from ~mesouser/Util directory to convert MM5 modeling system output to IEEE format data.

D.4.3 Accessing Source Code

Typically this is the way source code tar files are accessed in a (batch) Cray job deck:

```
#
#       Where is your source code?
#
#    set UseMySource = yes
     set UseMySource = no
#
#     ------ GET RAWINS TAR FILE ---------------------------
#
if ( $UseMySource == yes ) then
      rcp $Host/rawins.tar .
      tar -xf rawins.tar
else if ( ( $UseMySource == no ) && ( $ENVIRONMENT == BATCH ) ) then
      msread rawins.tar.Z /MESOUSER/MM5V2/RAWINS/RAWINS.TAR.Z
#     cp /fs/othrorgs/home0/mesouser/MM5V2/Rawins/rawins.tar.Z .
      uncompress rawins.tar
      tar -xf rawins.tar
endif
rm rawins.tar
```

As an example, the files contained in program RAWINS tar file is listed below:

CHANGES	Description of changes to the program
Diff/	Will contain difference files between consecutive releases
Makefile	Makefile to create the program executable
README	General information about the program directory
Templates/	Job deck directory
con.tbl	Table file for plots
configure.make.f77	Rules for the Makefile using cf77
configure.make.f90	Rules for the Makefile using f90
map.tbl	Table file for plots
src/	Program source code directory

D.4.4 Parameter Statements

The unix *cat* command is used to create FORTRAN include files, which are typically used to define the parameterized dimensions for a FORTRAN program. These are direct modifications to the source code, implying that strict FORTRAN syntax must be observed. Most programs require users to set up the parameter statements. The usage of *cat* is shown below:

```
cat > src/param.incl << EOF
      PARAMETER (.....)
      .....................
EOF
```

As an example, the following is taken from *datagrid.deck*:

```
cat > src/paramdim.incl << EOF
```

```
C
C   IMX AND JMX MUST BE GREATER THAN OR EQUAL TO THE MAXIMUM I AND J
C   DIMENSIONS OF ANY DOMAIN (EXPANDED OR NOT EXPANDED) THAT DATAGRID
C   WILL PROCESS.
C
      PARAMETER ( IMX=35, JMX=38 )
C
EOF
```

D.4.5 Fortran Namelist

The MM5 modeling system uses FORTRAN namelist to provide a way of selecting different options without re-compiling the program. The unix *cat* command is used to create namelist files from the shell script during the execution of the shell script. The variables in the namelist are described in detail in other chapters of this document specific to those individual programs. The format is the following

```
cat >! xxxx << EOF
&......
.........
& ;-----------------------------------
EOF
```

Since the namelist is not a ANSI 77 standard, the FORTRAN 77 compiler used by different machines may have different syntax for the namelist.

```
&DATIME ;---------DATE AND TIME INPUT--------------------------------
IYEAR  = 1993,                       ; Year (4-digit) of the start time.
 IMONTH = 03,                        ; Month of the start time.
 IDAY   = 13,                        ; Day of the start time.
 IHOUR  = 00,                        ; Hour of the start time.
 ;
 IFILES   = 2,                       ; NUMBER OF TIMES TO PROCESS
 ITIMINT  = 12,                      ; TIME INTERVAL (HOURS) (Usually 12 for
histor-
 ;                                   ; ical data. May be set to 6 for Unidata
input)
 ;
& ;---------------------------------------------------------------------------
EOF
#
###################################################################################
###################                                  ########################
###################      END USER MODIFICATION       ########################
###################                                  #########################
###############################################################################
```

After the user has 1) modified the QSUB commands to set the proper class for the job, 2) correctly set the shell variables for the switches and MSS pathnames, 3) modified the parameter statements, and 4) set up the FORTRAN namelist, there is typically no more user modification required in the deck. The rest of the script can be treated as a black box.

D.5 What is in the Remainder of a Job Deck?

D.5.1 Acquiring Input Files from the Mass Storage System

Syntax:

```
msread local-filename MSSfilename
```

where *local-filename* is the filename in the current directory, and *MSS filename* is the filename on MSS.

Example:

```
#
#       terrain files
#
set NUM = 1
while ( $NUM <= ${#InTerr} )
   if ( ! -e terrain.${NUM} ) then
      echo "  msread terrain.${NUM} $InTerr[$NUM] "
      msread terrain.${NUM} $InTerr[$NUM]
   endif
   @ NUM ++
end
```

msread:

This is a local NCAR command that copies an MSS file to UNICOS disk.

D.5.2 Setting Up Fortran Input and Output Units

Syntax:

```
assign -a filename [options] Fortran-unit
```

If no option is specified, the file must be Cray binary. If the file is 32-bit IEEE, the syntax becomes:

```
assign -a filename -Ff77 -Nieee Fortran-unit
```

Another way to setup input and output units is to use unix command *ln*:

```
ln -s filename Fortran-unit
```

This command makes a 'link' between filename and Fortran-unit.

Example:

```
#
#       set up fortran input files for DATAGRID
#
if ( -e assign.datgrd ) rm assign.datgrd
setenv FILENV assign.datgrd
if      (( $?SST ) && ( $SST == NAVY )) then
   assign -a navysst              fort.3
```

```
else if (( $?SST ) && ( $SST == CLIM )) then
   assign -a climsst              fort.12
endif
```

assign -a aliasfile:

This is a Cray utility that is used to assign FORTRAN files to FOTRAN units. The option -*a* expects actual file name to follow.

D.5.3 Creating FORTRAN Executable

Unix *make* utility is used to generate FORTRAN executable in the MM5 modeling system. The rules and compile options a *make* command uses are contained in the file: *configure.make* for programs TERRAIN, DATAGRID, RAWINS, INTERP and GRAPH, and *configure.user* for program MM5. For more information on make, please see Chapters 3 and 10 of this document.

D.5.4 Execution

```
    X.exe >&! X.print.out
```

Where X is the program name. Example:

```
#
#        run DATAGRID
#
date
if ( -e acct ) rm acct
datagrid.exe >&! datagrid.print.out
ja -s >! acct
cat acct >> datagrid.print.out
```

date:

This command prints the dates.

ja -s:

This UNICOS command ends the job accounting information. The -s option produces the summary report.

D.5.5 Save Files

When a job is completed, some files used in and output from the execution of a program are archived on MSS using the tar command:

```
tar -cf X.tar ......
```

An example:

```
tar -cf datagrid.out.tar src/datagrid.exe datagrid.namelist datagrid.print.out
```

D.5.6 Writing Output Files to Mass Storage System

Syntax:

```
mswrite -t $RetPd local-filename MSSfilename
```

Example:

```
set test = `ls -l rawanl.out`
   if ( $test[5] < 6000000000 ) then
       mswrite -t $RetPd rawanl.out ${OutRaw}/RAWINS_DOMAIN${DomId}
       echo mswrite -t $RetPd rawanl.out ${OutRaw}/RAWINS_DOMAIN${DomId}

   else
       bsplit -6000000000 rawanl.out  small.raw.
       set NUM = 1
       echo ls small.raw.* >! hold
       set test = `source hold`
       foreach split ( $test )
          mswrite -t $RetPd $split \
                  ${OutRaw}/RAWINS_DOMAIN${DomId}_BSPLIT_$LETTERS[$NUM]
          echo mswrite -t $RetPd $split \
                  ${OutRaw}/RAWINS_DOMAIN${DomId}_BSPLIT_$LETTERS[$NUM]
          @ NUM ++
       end
       rm small*
   endif
```

mswrite:

This is a NCAR local command used to copy an existing UNICOS disk file to MSS.

bsplit:

This is Cray executable used to split Cray binary files to a size specified (in this example, the size is 6,000,000,000 bytes). The size is limited by how many bytes MSS allows in a file (currently the limit is 6,000,000,000 bytes).

D.6 terrain.deck

```
# QSUB -r TERRAIN                        # request name
# QSUB -q reg                            # job queue class
# QSUB -eo                               # stdout and stderr together
# QSUB -lM 40Mw                          # maximum memory
# QSUB -lT 1000                          # time limit
# QSUB                                   # no more qsub commands
#
ja
set echo
#
#
```

```
#               ***************************************
#               ******   terrain batch C shell   ******
#               ******        V2 System          ******
#               ***************************************
#
if ( $?ENVIRONMENT ) then
   echo "environment variable defined as $ENVIRONMENT"
else
   setenv ENVIRONMENT INTERACTIVE
   echo "environment variable defined as $ENVIRONMENT"
endif
#
if ( $ENVIRONMENT == BATCH ) then
   cd $TMPDIR
endif
#
# This should be the user's case or experiment name.
#    set the TERRAIN MSS outputs pathname and the retention period
#    (for example MM5V2/TEST/EXP1/...).  Note that there is no / in front
#    of the pathname.
#
    set ExpName = MM5V2/TEST/EXP1
    set RetPd   = 365                   # Retention period in days for MSWRITEs
#
# set host user and machine names to remote copy the terrain plots
#    to users' local machine, where username is the users login name
#    on their local machine, domain.host is the local machine's address.
#
    set Host    = username@host.domain:/usr/tmp/username
#
# set GLOBAL30S = TRUE, if you want to use global 30 sec. elevation data:
#
#   set GLOBAL30S = TRUE
    set GLOBAL30S = FALSE
#
#   whether to create color-filled terrain/landuse plots
#
    set FillCo  = FALSE
#   set FillCo  = TRUE
#
#  set TerPlt = FALSE to create domain maps only
#  set TerPlt = TRUE  to create terrain/landuse output
#
    set TerPlt = FALSE
#   set TerPlt = TRUE
#
#  set NewLandUse = TRUE if you want to use a better landuse dataset
#      derived from 30 sec vegetation data
#
#  set NewLandUse = TRUE
   set NewLandUse = FALSE
#
#  if NewLandUse = TRUE, select which type of input data you would
#     like to use:
#     USGS = Global coverage, 24 catagories
#      SiB = 0 - 90N, 60W - 180W, 16 catagories (same as used in Eta)
#
if ( $NewLandUse == TRUE ) then
   set VegType = USGS
#  set VegType = SiB
endif
#
if ( $TerPlt == FALSE ) then
   set GLOBAL30S = FALSE
   set NewLandUse = FALSE
endif
#
#  TERRAIN MSS output file pathname
#
    set OutTerr = $ExpName
#
#  Do you want to create an additional IEEE copy of the output?
#     IEEE data are used on workstations.
#
```

```
    set OutIEEE = no
#   set OutIEEE = yes
#
#    ------ GET TERRAIN TAR FILE ---------------------------
#
msread terrain.tar.gz /MESOUSER/MM5V2/TERRAIN/NEW-TERRAIN.TAR.gz
gunzip -c terrain.tar | tar -xf -
rm terrain.tar.gz
#
#    ------ PARAMETER STATEMENTS -------------------------
#
cat >  src/parame.incl.tmp << EOF
C        IIMX,JJMX are the maximum size of the domains (can be
C          made larger). NSIZE = IIMX*JJMX
C
         PARAMETER (IIMX = 100, JJMX =100, NSIZE = IIMX*JJMX)
EOF
cat >  src/paramed.incl.tmp << EOF
C       ITRH,JTRH are the maximum size of the terrain data.
C       NOBT = ITRH*JTRH, here assuming
C        ITRH= 270 ( 45 deg. in north-south direction, 10 min. resolution)
C        JTRH= 450 ( 75 deg. in north-south direction, 10 min. resolution)
C       Use the second parameter statement if using 30 sec elevation data
        PARAMETER (ITRH = 500, JTRH = 500, NOBT = ITRH*JTRH)
C       PARAMETER (ITRH =1400, JTRH =1900, NOBT = ITRH*JTRH)
EOF
#
#    ------ TERRAIN NAMELISTS  ---------------------------
#    (Comments in the namelist are removed during execution)
#
if ( -e terrain.namelist ) rm terrain.namelist
cat > terrain.namelist << EOF
 &MAPBG
 PHIC   =   36.0,       ; CENTRAL LATITUDE (minus for southern hemesphere)
 XLONC =   -85.0,       ; CENTRAL LONGITUDE (minus for western hemesphere)
 IEXP   =    .T.,       ; .T. EXTENDED COARSE DOMAIN, .F. NOT EXTENDED.
 AEXP   =   360.,       ; APPROX EXPANSION(KM)
 IPROJ = 'LAMCON',      ; MAP PROJECTION
 ;IPROJ = 'POLSTR',     ; MAP PROJECTION
 ;IPROJ = 'MERCAT',     ; MAP PROJECTION
 & ;-----------------------------------------------------------------
 &DOMAINS
 ;
 MAXNES = 2,             ; max no of domains
 ;
 NESTIX=   35, 49,136,  1,  1,      ; DOMAIN  IX
 NESTJX=   41, 52,181,  1,  1,      ; DOMAIN  JX
 DIS    =   90.,30., 6., 1., 1.,    ; GRID SIZE(KM)
 MTHRD =     1,  1,  2,  1,  1,     ; MOTHER DOMAIN ID
 NESTI =     1, 10, 28,  1,  1,     ; LOWER LEFT I OF NEST
 NESTJ =     1, 17, 25,  1,  1,     ; LOWER LEFT J OF NEST
 RID    =  1.5,1.5,1.5,1.5,2.3,     ; RAD OF INFLUENCE IN GRID UNIT
 NTYPE =     2,  3,  5,  4,  4,     ; TERRAIN AND LANDUSE RESOLUTION
 ;
 ;  1:  1 deg (~111 km) global terrain and landuse
 ;  2: 30 min ( ~56 km) global terrain and landuse
 ;  3: 10 min ( ~19 km) global terrain and landuse
 ;  4;  5 min (  ~9 km) global terrain and landuse
 ;  5; 30 sec ( ~.9 km) global terrain and landuse
 ;            if NTYPE=5, set GLOBAL30S = T near the top of the deck
 ;
 NSTTYP=   1,   2,  2,  2,   2,    ;  1 -- ONE WAY NEST, 2 -- TWO WAY NEST
 & ;-----------------------------------------------------------------
 &OPTN
 IFTER    = .${TerPlt}.,  ; .T.-- TERRAIN,  .F.-- PLOT DOMAIN MAPS ONLY
 DATASW  = .T.,         ; .T. user specify terrain and landuse resolution (ntype)
 ;                        .F. terrain program choose the data resolution
 IFANAL  = .F.,         ; .T.-- OBJECTIVE ANALYSIS, .F.-- INTERPOLATION
 ISMTHTR =  1 ,         ;  1: 1-2-1 smoother, 2: two pass smoother/desmoother
 IFEZFUG = .T.,         ; .T. USE EZMAP WATER BODY INFO TO FUDGE THE LAND USE
 IFTFUG  = .F.,         ; .T. DON'T DO EZFUDGE WITHIN THE USER-SPECIFIED
 ;                        LAT/LON BOXES, need to define namelist fudget
 IFFUDG  = .F.,         ; .T. POINT-BY-POINT FUDGING OF LANDUSE,
                        ;  need to define namelist fudge
```

```
 IPRNTD   = .F.,        ; PRINT OUT LAT. AND LON. ON THE MESH
 IPRTHT   = .F.,        ; PRINT OUT TERRAIN HEIGHT ON THE MESH
 IPRTLU   = .F.,        ; PRINT OUT LANDUSE ON THE MESH
 FIN     = 300.,300.,300.,100.,100., ; CONTOUR INTERVAL (meter) FOR TERRAIN HEIGHT PLOT
 ;-------------------------------------------------------------------------
 ;If your true latitude is not standard for the map projection you choose
 ;uncomment and define the TRUELAT1 and TRUELAT2 lines (delete the ;).
 ;standard (minus for southern hemesphere) :
 ;          Lambert Conformal   : TRUELAT1=30. ,TRUELAT2=60.,
 ;          Polar Stereographic: TRUELAT1=60.,
 ;          Mercater            : TRUELAT1=0.               )
 ;TRUELAT1=91.,         ; TRUE LATITUDE 1
 ;TRUELAT2=91.,         ; TRUE LATITUDE 2, use this if IPROJ='LAMCON'
 IFILL     = .$FillCo.,         ; COLOR-FILL THE TERRAIN HEIGHT CONTOURS AND MAP BKG
 VegeSoil = .${NewLandUse}.,    ; .TRUE. --- Use vegetation for landuse
  VegOnly = .${NewLandUse}.,    ; .TRUE. --- Use vegetation for landuse
 G30SD     = .${GLOBAL30S}.,    ; .TRUE. --- Use global 30 sec elevation data
 ;                     (do not change)
 & ;-------------------------------------------------------------------
 &FUDGE
 ; USE ONLY IF IFFUDG = .T., POINT-BY-POINT FUDGING OF LANDUSE,
 ; IFFUG FOR EACH OF THE NESTS: .F. NO FUDGING, .T. FUDGING
 IFFUG    = .F.,.F.,    ; FUDGE FLAGS
 ; NDFUG : THE NUMBER OF FUDGING POINTS FOR EACH OF NESTS
 NDFUG    = 0,0,
 ; LOCATION (I,J) AND LANDUSE VALUES FOR EACH OF THE NESTS
 ; NOTE: REGARDLESS OF IFFUG AND NDFUG, 200 VALUES MUST BE GIVEN FOR
 ;       EACH NEST, OR ELSE THE INDEXING WILL GET MESSED UP
 ; The example below is for two domains. Add more for domain 3 and up
 ;       if needed. Do not remove 0 values for domain 1 and/or 2 even
 ;       they are not used.
 ;
 IFUG(1,1)=  200*0,     ; I location for fudge points in domain 1
 IFUG(1,2)=  200*0,     ; I location for fudge points in domain 2
 JFUG(1,1)=  200*0,     ; J location for fudge points in domain 1
 JFUG(1,2)=  200*0,     ; J location for fudge points in domain 2
 LNDFUG(1,1)= 200*0,    ; land-use value at fudge points for domain 1
 LNDFUG(1,2)= 200*0,    ; land-use value at fudge points for domain 2
 & ;-------------------------------------------------------------------
 &FUDGET
 ; USE ONLY IF IFTFUG=.T., WHICH MEANS TERRAIN WON'T DO EZFUDGE WITHIN
 ;    THE USER-SPECIFIED LAT/LON BOXES.  THIS OPTION IS USED WHEN THERE
 ;    ARE INLAND BODIES OF WATER THAT ARE DEFINED IN THE LAND USE
 ;    DATA SET BUT NOT IN THE EZMAP DATA SET.  THIS OPTION PREVENTS
 ;    THOSE BODIES OF WATER FROM BEING WIPED OUT BY EZFUDGE
 NFUGBOX = 2              ;  NUMBER OF SUBDOMAINS IN WHICH TO
 ;                           TURN OFF EZMAP LAND USE FUDGING
 STARTLAT=45.0,44.0,      ;  LATITUDES OF LOWER-LEFT CORNERS OF SUBDOMAINS
 ENDLAT  =46.5,45.0,      ;  LATITUDES OF UPPER-RIGHT CORNERS OF SUBDOMAINS
 STARTLON=-95.0,-79.8,    ;  LONGITUDES OF LOWER-LEFT CORNERS OF SUBDOMAINS
 ENDLON  =-92.6,-78.5,    ;  LONGITUDES OF UPPER-RIGHT CORNERS OF SUBDOMAINS
 & ;-------------------------------------------------------------------
 &EZFUDGE
 ; USE ONLY IF IFEZFUG=.T., WHICH TURNS ON EZMAP WATER BODY FUDGING OF LANDUSE.
 ; USERS: FEEL FREE TO ADD ANY MORE LAKE SURFACE HEIGHTS THAT YOU'LL NEED.
 ; HTPS IS THE HEIGHT IN METERS AND THE INDEX OF HTPS CORRESPONDS TO THE ID
 ; OF THE `PS' AREA IN THE FILE ezmap_area_ids.
 ;
 HTPS(441) =  -.001   ; Oceans  --  Do NOT change this one
 HTPS(550) =  183.    ; Lake Superior
 HTPS(587) =  177.    ; Lakes Michigan and Huron
 HTPS(618) =  176.    ; Lake St. Clair
 HTPS(613) =  174.    ; Lake Erie
 HTPS(645) =   75.    ; Lake Ontario
 HTPS(480) = 1897.    ; Lake Tahoe
 HTPS(500) = 1281.    ; Great Salt Lake
 & ;-------------------------------------------------------------------
EOF
#
#########################################################################
####################                             #######################
####################    END USER MODIFICATION    #######################
####################                             #######################
#########################################################################
```

```
#
#
#    This is where the standard source code/executable come from
#
set MesoUser = /fs/othrorgs/home0/mesouser/Util
#
#    The make rules are contained in the file configure.make
#
# --------- create terrain executable -------------------
#
   set DomId = (1 2 3 4 5 6 7 8 9 10)
#
# check to see if parameter statements are different
#
cd src
../Templates/incldiff.sh parame.incl.tmp parame.incl
../Templates/incldiff.sh paramed.incl.tmp paramed.incl
cd ..
#
echo "using configure.make file with f90 option"
cp configure.make.CRAY configure.make
#
# edit namelist to remove comments
#
sed -f ./Templates/no_comment.sed terrain.namelist > mif.tmp
mv mif.tmp terrain.namelist
#
# --------- get terrain and landuse data set from MS
#
cd Data
if (( ! -e landu.30 ) && ( ${TerPlt} == TRUE ) ) then
   msread terln.tar /MESOUSER/MM5V2/TERRAIN/terln.tar
   tar -xf terln.tar
endif
#
#      Get vegetation/land-water mask data if NewLandUse = TRUE
#
if ( $NewLandUse == TRUE ) then

  if ( $VegType == USGS ) then

#
# -- 60, 30, 10, and 5 min Land-Water mask and vegetation data:
#
    if ( ! -e lwmask-usgs.30 ) then
        msread landwatermask-usgs.tar \
               /MESOUSER/MM5V2/TERRAIN/DATA/landwatermask-usgs.tar
        tar -xf landwatermask-usgs.tar
        rm landwatermask-usgs.tar
    endif
    if ( ! -e veg-usgs.30 ) then
        msread vegetation-usgs.tar \
               /MESOUSER/MM5V2/TERRAIN/DATA/vegetation-usgs.tar
        tar -xf vegetation-usgs.tar
        rm vegetation-usgs.tar
    endif
#
# -- 30 sec Land-Water mask and vegetation data:
#
    if ( $GLOBAL30S == TRUE ) then
    if ( ! -e lwmask-usgs.30s ) then
       msread lwmask-usgs.30s /MESOUSER/MM5V2/TERRAIN/DATA/landwatermask30s-usgs
    endif
    if ( ! -e veg-usgs.30s ) then
       msread veg-usgs.30s /MESOUSER/MM5V2/TERRAIN/DATA/vegetation30s-usgs
    endif
    endif

  else if ( $VegType == SiB ) then
#
# -- Land-Water mask data:
#
    if ( ! -e lwmask-sib.30 ) then
       msread landwatermask-sib.tar /MESOUSER/MM5V2/TERRAIN/DATA/landwatermask-sib.tar
```

```
       tar -xf landwatermask-sib.tar
       rm landwatermask-sib.tar
    endif
# -- Vegetation data:
    if ( ! -e vegetation-sib.tar ) then
       msread vegetation-sib.tar /MESOUSER/MM5V2/TERRAIN/DATA/vegetation-sib.tar
       tar -xf vegetation-sib.tar
       rm vegetation-sib.tar
    endif

  endif

endif
cd ..
#
#      Set local file names
#
if ( $NewLandUse == TRUE ) then

   if ( $VegType == USGS ) then

      set LwmData60m = Data/lwmask-usgs.60
      set LwmData30m = Data/lwmask-usgs.30
      set LwmData10m = Data/lwmask-usgs.10
      set LwmData05m = Data/lwmask-usgs.05
      set VegData60m = Data/veg-usgs.60
      set VegData30m = Data/veg-usgs.30
      set VegData10m = Data/veg-usgs.10
      set VegData05m = Data/veg-usgs.05
      if ( $GLOBAL30S == TRUE ) then
         set LwmData30s = Data/lwmask-usgs.30s
         set VegData30s = Data/veg-usgs.30s
      endif
      cp src/paramesv1.incl src/paramesv.incl.tmp
      cp src/vs_data2.incl  src/vs_data.incl.tmp
      ./Templates/incldiff.sh src/paramesv.incl.tmp src/paramesv.incl
      ./Templates/incldiff.sh src/vs_data.incl.tmp src/vs_data.incl

   else if ( $VegType == SiB ) then

      set LwmData60m = Data/lwmask-sib.60
      set LwmData30m = Data/lwmask-sib.30
      set LwmData10m = Data/lwmask-sib.10
      set LwmData05m = Data/lwmask-sib.05
      set LwmData30s = Data/lwmask-sib.30s
      set VegData60m = Data/veg-sib.60
      set VegData30m = Data/veg-sib.30
      set VegData10m = Data/veg-sib.10
      set VegData05m = Data/veg-sib.05
      set VegData30s = Data/veg-sib.30s
      cp src/paramesv0.incl src/paramesv.incl.tmp
      cp src/vs_data0.incl  src/vs_data.incl.tmp
      ./Templates/incldiff.sh src/paramesv.incl.tmp src/paramesv.incl
      ./Templates/incldiff.sh src/vs_data.incl.tmp src/vs_data.incl

   else
      echo `STOP due to wrong VegType ` $VegType
      exit (1)
   endif
endif
#
# --------- set up fortran input files for TERRAIN ----------------
#
set ForUnit = fort.
rm ${ForUnit}9 ${ForUnit}1* ${ForUnit}2* ${ForUnit}4*
#
   ln -s  namelist              ${ForUnit}9
   ln -s  terrain.namelist      ${ForUnit}15
   ln -s  ezids                 ${ForUnit}18
#
   assign -a  raobsta.ieee    -Ff77  -Nieee  ${ForUnit}16
   assign -a  Data/landu.60   -Ff77  -Nieee  ${ForUnit}20
   assign -a  Data/ter.60     -Ff77  -Nieee  ${ForUnit}21
   assign -a  Data/landu.30   -Ff77  -Nieee  ${ForUnit}22
```

```
   assign -a  Data/ter.30     -Ff77  -Nieee  ${ForUnit}23
   assign -a  Data/landu.10   -Ff77  -Nieee  ${ForUnit}24
   assign -a  Data/ter.10     -Ff77  -Nieee  ${ForUnit}25
   assign -a  Data/ter.05     -Ff77  -Nieee  ${ForUnit}27

   if (   $NewLandUse == TRUE ) then
      assign -a $LwmData60m   -Ff77  -Nieee  ${ForUnit}40
      assign -a $LwmData30m   -Ff77  -Nieee  ${ForUnit}41
      assign -a $LwmData10m   -Ff77  -Nieee  ${ForUnit}42
      assign -a $LwmData05m   -Ff77  -Nieee  ${ForUnit}43
#
      assign -a $VegData60m   -Ff77  -Nieee  ${ForUnit}45
      assign -a $VegData30m   -Ff77  -Nieee  ${ForUnit}46
      assign -a $VegData10m   -Ff77  -Nieee  ${ForUnit}47
      assign -a $VegData05m   -Ff77  -Nieee  ${ForUnit}48
#
      if ( $GLOBAL30S == TRUE ) then
         assign -a $LwmData30s   -Ff77  -Nieee  ${ForUnit}44
         assign -a $VegData30s   -Ff77  -Nieee  ${ForUnit}49
      endif
   endif
#
#  Obtain 30 sec global elevation data and preprocess the data
#
if ( $GLOBAL30S == TRUE ) then
#
# (1) Generate the "dem_read" for getting the 30s data:
#
      make data_area.exe
      src/data_area.exe > data_area.out
      set toast = $status
      if ( $toast != 0 ) then
         echo "error in running data_area, stopping"
         exit(1)
      endif

# only if dem_read is created, you need to go through the following sections

  if ( -e dem_read ) then

      echo "---------------------------------------------------"
      echo "Global 30-sec data files needed"
      echo "   FileName     MSS_file      FTP_file    FortranUnit"
      cat dem_read
      echo "---------------------------------------------------"
#
# (2) Get the data from MSS:
#
      mkdir -p Data30s
      cd Data30s ; mv ../dem_read .
        set File30s = `cat dem_read`
        set Nfiles = ${#File30s}
        echo $Nfiles $File30s
#
        set Num0 = 1
        while ( $Num0 <= $Nfiles )
          @ Num1 = $Num0 + 1
          @ Num2 = $Num0 + 2
          @ Num3 = $Num0 + 3

            if  ( ! -e $File30s[$Num0] ) msread $File30s[$Num0] $File30s[$Num1]
            if  ( -e ${ForUnit}$File30s[$Num3] ) rm ${ForUnit}$File30s[$Num3]
            assign -a $File30s[$Num0]  -Ff77 -Nieee ${ForUnit}$File30s[$Num3]

          @ Num0 = $Num0 + 4
        end
      cd ..
#
# (3) To reconstruct the data for use of TERRAIN:
#  .. mv some files to directory: Data30s
#
      echo "beginning reconstruct data"
      mv data30sID  Data30s/.
      mv para.incl  src/.
```

```
      make rdem.exe
      mv src/rdem.exe Data30s/.
      cd Data30s
        rm new_*
        rdem.exe > rdem.out
        set toast = $status
        if ( $toast != 0 ) then
           echo "error in running rdem, stopping"
           exit(3)
        endif
        mv new_* ../Data/.
        rm gmeta rdem.exe data30sID ${ForUnit}79 ${ForUnit}8*
      cd ..
  else
    echo "30 SEC ELEVATION DATA WERE NOT REQUESTED. CHECK NTYPE IN NAMELIST"
    set GLOBAL30S = FALSE
  endif

endif
#
if ( $GLOBAL30S == FALSE ) then
# ....... use 48 US 30 sec data:
     assign -a  Data/ter.30s    -Ff77  -Nieee  ${ForUnit}29
endif
#
#       make TERRAIN executable and run TERRAIN
#
make terrain.exe
#
if ( ! -e terrain.exe ) ln -s src/terrain.exe  terrain.exe
#
date
if ( -e acct ) rm acct
chmod +x terrain.exe
terrain.exe >&! terrain.print.out
#
ja -s >! acct
cat acct >> terrain.print.out
if ( $ENVIRONMENT == BATCH ) cat terrain.print.out
#
set toast = $status
if (( $ENVIRONMENT == BATCH ) && ( $toast != 0 )) then
   echo "error in the TERRAIN, stopping"
   rm core
   exit(1)
endif
#
#       if interactive, probably do not want to zap and dispose
#
if ( $ENVIRONMENT != BATCH ) then
   echo -n "test terrain run complete, continue?     (no)    "
   set ans = "$<"
   if (( $ans != "y" ) && ( $ans != "yes" )) then
      exit (0)
   endif
endif
#
rm ${ForUnit}*
#
# send terrain plots back to the front end machine
#
rcp TER.PLT ${Host}/ter.plt
#
if ( $TerPlt == TRUE || $TerPlt == T ) then
#
#       terrain output files to mss
#
set Numb = 0
set IfAny = ( `ls TERRAIN_DOMAIN*` )
set MAX = ${#IfAny}
if ( $MAX > 0 ) then
   foreach fil ( `ls TERRAIN_DOMAIN*` )
   @ Numb ++
      echo "mswrite -t $RetPd $fil $OutTerr/TERRAIN_DOMAIN${DomId[$Numb]}"
```

```
      mswrite -t $RetPd $fil $OutTerr/TERRAIN_DOMAIN${DomId[$Numb]}
      if ( $OutIEEE == yes ) then
         if ( ! -e ieee.csh ) then
            cp $MesoUser/ieee.csh .
            chmod +x ieee.csh
         endif
         ieee.csh $fil
         mswrite -t $RetPd ${fil}.ieee $OutTerr/TERRAIN_DOMAIN${DomId[$Numb]}_IEEE
      endif
   end
endif
#
#  send terrain plots to various plotting device
#
echo " mswrite -t $RetPd TER.PLT $OutTerr/TERRAIN_PLOT "
mswrite -t $RetPd TER.PLT $OutTerr/TERRAIN_PLOT
#
endif
#
# tar the namelist, source code, executable, and output together
#     and save the tar file on MSS
#
tar -cf terrain.out.tar terrain.namelist terrain.print.out \
    src/terrain.exe
#
echo " mswrite -t $RetPd terrain.tar $ExpName/terrain.out.tar "
mswrite -t $RetPd terrain.out.tar $OutTerr/terrain.out.tar
rm G*
ls -ls
#
ja -st
```

D.7 regrid.deck

```
# QSUB -r RGD                           # request name
# QSUB -q reg                           # job queue class
# QSUB -eo                              # stdout and stderr together
# QSUB -lM 6Mw                          # maximum memory
# QSUB -lT 1800                         # time limit
# QSUB -mb                              # mail me when it starts
# QSUB -me                              # mail me when it ends
# QSUB                                  # no more qsub commands
#
###############################################################################
###############################################################################
#####                                                                    #####
#####   This script is a CRAY job deck for the REGRID package.  It is    #####
#####   designed for submittal to NCAR's CRAY machines.  If you are not  #####
#####   working with NCAR's CRAY machines, this script will not be of    #####
#####   much help to you.                                                #####
#####                                                                    #####
#####   This script is set up for only a few specific data sets that are #####
#####   archived at NCAR and easily accessible from NCAR's CRAY machines. #####
#####   If you want to use other data sets, you will likely need to set  #####
#####   up the "pregrid" and "regridder" portions of REGRID yourself, and #####
#####   run interactively.                                               #####
#####                                                                    #####
#####   This deck performs just the basics.  It represents a subset of   #####
#####   the full flexibility and capability of the REGRID package.  For  #####
#####   more control, investigate interactive use.                       #####
#####                                                                    #####
#####   This deck is still in the experimental phase.  Capabilities and  #####
#####   options are likely to change with unusual stealth and alarming   #####
#####   frequency.  Please be patient.                                   #####
#####                                                                    #####
#####   Questions, comments, and lavish praise should be directed to:    #####
#####                        mesouser@ucar.edu                           #####
#####                                                                    #####
###############################################################################
###############################################################################
```

```
ja

set echo

cd $TMPDIR
#
# Select the source of first-guess analyses from among the following
# datasets archived at NCAR:
#

#   set ARCHIVE = GDAS    # NCEP Global Data Assimilation System,
                           # archives begin 1997-04-01

#   set ARCHIVE = ON84     # NCEP GDAS Archives through 1997-03-31

    set ARCHIVE = NNRP     # NCEP/NCAR Reanalysis Project

cat << End_Of_Record2 | sed -e 's/#.*//; s/  *$//' > ./pregrid.namelist
&record1
#
# Set the starting date of the time period you want to process:
#
 START_YEAR  = 1993    # Year (Four digits)
 START_MONTH = 03      # Month ( 01 - 12 )
 START_DAY   = 12      # Day ( 01 - 31 )
 START_HOUR  = 00      # Hour ( 00 - 23 )

 END_YEAR  = 1993      # Year (Four digits)
 END_MONTH = 03        # Month ( 01 - 12 )
 END_DAY   = 15        # Day ( 01 - 31 )
 END_HOUR  = 00        # Hour ( 00 - 23 )
#
# Define the time interval to process.
#
 INTERVAL =  43200 # Time interval (seconds) to process.
       # This is most sanely the same as the time interval for
                   # which the analyses were archived, but you can really
                   # set this to just about anything, and pregrid will
                   # interpolate in time and/or skip over time periods for
                   # your regridding pleasure.
/

&record2
#
# Set parameters related to pressure levels.  Units are Pascals (Pa).
#
 PTOP_IN_PA         = 10000        # Top pressure level to extract

 NEW_LEVELS_IN_PA   =  97500, 95000, 92500, 90000,  # Extra levels to
                       87500, 85000, 82500, 80000,  # which regridder
                       77500, 75000, 72500, 70000,  # is to interpolate
                       67500, 65000, 62500, 60000,
                       57500, 55000, 52500, 50000,
                       47500, 45000, 42500, 40000,
                       37500, 35000, 32500, 30000,
                       27500, 25000, 22500, 20000,
                       17500, 15000, 12500, 10000,
 /
End_Of_Record2

#######  END OF NAMELIST OPTIONS. ########

#
# Set your terrain file name(s):
#
   set InTerr = /KMANNING/TERRAIN/US60/EXPANDED/TERRAIN_DOMAIN1

#
# Output Options:
#

# Do you convert to Old-format datagrid files (i.e., do you run pablum?)
# If you don't want old-format files, don't set this variable.
```

```
#     set OldFormat

# Where to write on the MSS.  If you don't want to write the data to MSS,
# don't set MssPath.

#      set MssPath = REGRID/TEST

# Set this to get 32-bit IEEE output.  If you want the CRAY-Binary files,
# don't set IEEE32.

      set IEEE32
#
#  Where to send the data via rcp.  If you don't want to send the data,
#  don't set SEND_LOCAL:

      set SEND_LOCAL = wahoo2.mmm:/mmmtmp/kmanning

################################################################################
################################################################################
##########                                                          ##########
##########                   END USER MODIFICATION                  ##########
##########                                                          ##########
################################################################################
################################################################################

#
# Get the REGRID tarfile:
#

cat >! ftp.cmd << EOF0
open ftp.ucar.edu
user anonymous ${user}@ucar.edu
cd mesouser/newprogs
get regrid.tar.gz
EOF0

ftp -n < ftp.cmd
rm ftp.cmd

gzip -d regrid.tar.gz
pax -rf regrid.tar

cd REGRID
mv ../pregrid.namelist .
set TOPDIR = `pwd`

cd pregrid
    cat ${TOPDIR}/pregrid.namelist

##############################################################################
#
# Get the data:
   f90 -f free -o rdnml_util util/rdnml_util.F
   set DtLst = `rdnml_util < ${TOPDIR}/pregrid.namelist`

   set START_DATE = ${DtLst[1]}-${DtLst[2]}-${DtLst[3]}_${DtLst[4]}
   set END_DATE = ${DtLst[5]}-${DtLst[6]}-${DtLst[7]}_${DtLst[8]}

if ( $ARCHIVE == NNRP ) then
   cd ${TOPDIR}/pregrid/nnrp
   chmod +x get_nnrp.csh
   get_nnrp.csh ${START_DATE} ${END_DATE} 6
else if ( $ARCHIVE == GDAS ) then
   cd ${TOPDIR}/pregrid/ncep.grib
   chmod +x get_ncep.csh
   get_ncep.csh ${START_DATE} ${END_DATE} 12
else if ( $ARCHIVE == ON84 ) then
   cd ${TOPDIR}/pregrid/on84
   chmod +x get_on84.csh
   get_on84.csh ${START_DATE} ${END_DATE} 12
else
   exit(2)
endif
```

```
############################################################################

# Run pregrid

  cd ${TOPDIR}
  make cray >&! make.print.out

  cd pregrid

  set LETTERS = ( A B C D E F G H I J K L M N O P Q R S T U V W X Y Z )

if ( $ARCHIVE == NNRP ) then
     cd grib.misc
     rm -f FILE:*
     rm -f GRIBFILE*

     ln -s ${TOPDIR}/pregrid.namelist pregrid.namelist

     if ( -e Vtable ) then
        rm Vtable
     endif
     touch Vtable
     cat Vtable.NNRP >> Vtable

     set NUM = 0
     set num = 1
     foreach file ( ../nnrp/pgb* ../nnrp/SFC* )
        @ NUM ++
        if ( $NUM == 27 ) then
           set NUM = 1
           @ num ++
        endif
        ln -s ${file} GRIBFILE.${LETTERS[$num]}${LETTERS[$NUM]}
     end

     pregrid_grib.exe

     set ROOT = `pwd`
     set ROOT = ( \'${ROOT}/FILE\' )

else if ( $ARCHIVE == GDAS ) then

    cd ${TOPDIR}/pregrid/ncep.grib
    rm GRIBFILE*
    # Build the Vtable
    rm -f Vtable
    cp Vtable.NCEP Vtable
    cat Vtable.SST >> Vtable
    cat Vtable.SNOW >> Vtable

    # Get the input files
    ln -s ${TOPDIR}/pregrid.namelist pregrid.namelist

    cat pregrid.namelist

    set Num = 0
    foreach file ( NCEP_GRIB* )
       @ Num ++
       ln -s $file GRIBFILE${LETTERS[$Num]}
    end

    # Run the program
    pregrid_ncep.exe

    set ROOT = `pwd`
    set ROOT = ( \'${ROOT}/SST\', \'${ROOT}/NCEP\', \'${ROOT}/SNOW\')
    ls -l

else if ( $ARCHIVE == ON84 ) then

    cd ${TOPDIR}/pregrid/on84
    ln -s ${TOPDIR}/pregrid.namelist pregrid.namelist
```

```
    # Build the Vtable
    rm -f Vtable
    touch Vtable
    cat Vtable.ON84 >> Vtable
    cat Vtable.SST >> Vtable
    cat Vtable.SNOW >> Vtable
    set Num = 0
    # Get the input files
    foreach file ( ./NCEP_ON84* )
       @ Num ++
       ln -s $file ON84FILE${LETTERS[$Num]}
    end

    # Run the program
    pregrid_on84.exe

    # Set up the ROOT variable.

    set ROOT = `pwd`
    set ROOT = ( \'${ROOT}/ON84\', \'${ROOT}/SST\', \'${ROOT}/SNOW\')

endif

###############################################################################

cd ${TOPDIR}/regridder

msread terrain ${InTerr}

rm -f namelist.input
cp ${TOPDIR}/pregrid.namelist namelist.input
###cat << End_Of_Namelist | sed -e 's/#.*//; s/  *$//' >> ./namelist.input
cat << End_Of_One >> ./namelist.input

&record3
 root       = $ROOT[1]
End_Of_One

set n = 1
while ( $n < $#ROOT )
   @ n++
cat << End_Of_Next >> ./namelist.input
             $ROOT[$n]
End_Of_Next
end

cat << End_Of_Namelist >> ./namelist.input

 terrain_file_name              = './terrain' /

&record4
 print_echo                     = .FALSE. ,
 print_debug                    = .FALSE. ,
 print_mask                     = .FALSE. ,
 print_interp                   = .FALSE. ,
 print_link_list_store          = .FALSE. ,
 print_array_store              = .FALSE. ,
 print_header                   = .FALSE. ,
 print_output                   = .FALSE. ,
 print_file                     = .FALSE. /
End_Of_Namelist

cat namelist.input

regridder

if ($?OldFormat) then
   pablum
endif

if ( $?IEEE32 ) then
   /fs/othrorgs/home0/mesouser/Util/ieee.csh datagrid
   if ($?OldFormat) then
      /fs/othrorgs/home0/mesouser/Util/ieee.csh datagrid_old_format
```

```
   endif
endif

if ( $?MssPath ) then
   if ( $?IEEE32 ) then
      if ($?OldFormat) then
         mswrite datagrid_old_format.ieee ${MssPath}/DATAGRID_OLD_FORMAT_IEEE
      else
         mswrite datagrid.ieee ${MssPath}/DATAGRID_IEEE
      endif
   else
      if ($?OldFormat) then
         mswrite datagrid_old_format ${MssPath}/DATAGRID_OLD_FORMAT
      else
         mswrite datagrid ${MssPath}/DATAGRID
      endif
   endif
endif

if ( $?SEND_LOCAL ) then
   if ( $?IEEE32 ) then
      if ($?OldFormat) then
         rcp datagrid_old_format.ieee ${SEND_LOCAL}/datagrid_old_format
      else
         rcp datagrid.ieee ${SEND_LOCAL}/datagrid
      endif
   else
      if ($?OldFormat) then
         rcp datagrid_old_format ${SEND_LOCAL}
      else
         rcp datagrid ${SEND_LOCAL}
      endif
   endif
endif

ls

ja -st
```

D.8 rawins.deck

```
# QSUB -r RAWINS                       # request name
# QSUB -q reg                           # job queue class
# QSUB -eo                              # stdout and stderr together
# QSUB -lM 8Mw                         # maximum memory
# QSUB -lT 900                          # time limit
# QSUB -lt 900
# QSUB                                  # no more qsub commands
#
ja
set echo
#
#               ***************************************
#               *******   rawins batch C shell   *******
#               *******       V2 System          *******
#               *******   for both cf77 and f90  *******
#               ***************************************
#
#     This version of RAWINS job deck can only process one analysis domain
#     per RAWINS submittal.  The nested domain center latitude and longitude
#     can be different from the coarse domain.
#     For example, if one wants to obtain analysis on domain 1 and 2, then
#     RAWINS needs to be run twice for domian 1 and domain 2 respectively.
#
#
```

```
if ( $?ENVIRONMENT ) then
   echo "environment variable defined as $ENVIRONMENT"
else
   setenv ENVIRONMENT INTERACTIVE
   echo "environment variable defined as $ENVIRONMENT"
endif
#
if ( $ENVIRONMENT == BATCH ) then
   cd $TMPDIR
endif
#
#       this should be the user's case or experiment
#
   set ExpName = MM5V2/TEST/EXP1    # RAWINS input/output MSS pathname
   set RetPd   = 365                # Retention period for MSWRITEs
#
#       type of rawins job
#
   set Submit  = 0           # 0 = no autobogus; 1 = autobogus 1; 2 = autobogus 2
#
   set INOBS   = ARCHIVE
#  set INOBS   = UNIOBS
#
#  set SFCsw   = NoSFC
   set SFCsw   = SFC
#
   set BOGUSsw = NoBOG
#  set BOGUSsw = Konly
#  set BOGUSsw = Nonly
#  set BOGUSsw = KandN
#
#       locations for datasets
#
#   Domain id for RAWINS to process (RAWINS does only one domain per submittal)
#
   set DomId  = 1
#
   set InDatg  =  ${ExpName}/DATAGRID_DOMAIN${DomId}
#
#
#       Find MSS names for observations from the following catalogs:
#            upper air: /fs/othrorgs/home0/mesouser/catalog/catalog.raob
#              surface: /fs/othrorgs/home0/mesouser/catalog/catalog.sfc
#
if ( $INOBS == ARCHIVE ) then
   set IfUni = F
   set InRaobs   =  ( /DSS/Y16320 /DSS/Y16321 )
   if ( $SFCsw  == SFC ) then
      set InSfc6h   =  ( /DSS/Y16310 /DSS/Y16311 )   # see catalog.sfc, list A
      set InSfc3h   =  ( /DSS/Y16315 /DSS/Y16316 )   # see catalog.sfc, list B
   endif
else
   set IfUni = T
   set upa_unidate = ( xxxxxxxx )                 # YYMMDDHH
   set sfc_unidate = ( xxxxxxxx )                 # YYMMDDHH
endif
#
#   RAWINS MSS OUTPUT FILE NAMES
#
set OutRaw    =   ${ExpName}
```

```
#
#  Do you want to create an additional IEEE copy of the output?
#     IEEE data are used on workstations.
#
    set OutIEEE = no
#   set OutIEEE = yes
#
#  set up the remote host location where your source code may locate
#
   set Host    = username@host.domain:/usr/tmp/username
#
#      Do you want to use your source code tar file?
#
#  set UseMySource = yes
   set UseMySource = no
#
#    ------ GET RAWINS TAR FILE ---------------------------
#
if ( $UseMySource == yes ) then
     rcp $Host/rawins.tar .
     tar -xf rawins.tar
else if ( ( $UseMySource == no ) && ( $ENVIRONMENT == BATCH ) ) then
     msread rawins.tar.Z /MESOUSER/MM5V2/RAWINS/RAWINS.TAR.Z
#    cp /fs/othrorgs/home0/mesouser/MM5V2/Rawins/rawins.tar.Z .
     uncompress rawins.tar
     tar -xf rawins.tar
endif
rm rawins.tar
#
#    ------ RAWINS PARAMETER STATEMENT ---------------------------
#
if ( -e src/paramdim.incl ) rm src/paramdim.incl
cat > src/paramdim.incl << EOF
C
C  IMX,JMX, MUST CORRESPOND TO THE DIMENSIONS IN THE INPUT FILE.  THESE
C  WILL BE THE EXPANDED DIMENSIONS IF THIS IS THE COARSE GRID, AND THE
C  EXPANDED OPTION WAS SELECTED.
C
C  LMX MUST BE GREATER THAN OR EQUAL TO THE MAXIMUM NUMBER OF LEVELS
C  (PRESSURE LEVELS + SURFACE).
C
      PARAMETER(IMX=45,JMX=51,LMX=21)
C------------------------------------------------------------------
EOF
#
#       parameters for rawins, user adjustable for core constraints
#
if (-e src/paramirb.incl ) rm src/paramirb.incl
cat > src/paramirb.incl << EOF
C-----------------------------------------------------------------------
C   IRB MUST BE AT LEAST THE MAXIMUM NUMBER OF RAOB STATIONS
C
      PARAMETER(IRB=500)
EOF
if (-e src/paramirs.incl ) rm src/paramirs.incl
cat > src/paramirs.incl << EOF
C-----------------------------------------------------------------------
C   IRS MUST BE AT LEAST THE MAXIMUM NUMBER OF SFC. + RAOB STATIONS
C
      PARAMETER(IRS=3000)
```

```
EOF
#
#    ------ RAWINS LOCAL NAMELIST ----------------------------------
#          (you need not change anything of the form: ${____})
#          (Comments in the namelist are removed during execution)
#
if ( -e rawins.namelist ) rm rawins.namelist
cat > rawins.namelist << EOF
 &LOCMIF ;---------LOCAL NAMELIST FOR PROGRAM RAWINS--------------------
 NNEWPL = 10, ;  NUMBER OF NEW PRESSURE LEVELS
 GNLVL = 950.,925.,900.,800.,750.,650.,600.,550.,450.,350.,
 ;                    PRESSURES AT THE NEW PRESSURE LEVELS (BOTTOM TO TOP)
 ;
 ;----------OBJECTIVE ANALYSES OPTIONS FOLLOW--------------------------
 IWTSCM = 3,        ; 1:  CIRCULAR WEIGHTING FUNCTION (CRESSMAN)
 ;                  ; 2:  ELLIPTICAL WEIGHTING FUNCTION
 ;                  ; 3:  BANANA-SHAPED WEIGHTING FUNCTION
 ;                  ; 4:  MULTIQUADRIC INTERPOLATION
 IFAC = 1,          ; 1:  FIRST-GUESS FIELD EXISTS, RAWINS ALTERS IT.
 ;                  ; 2:  FIRST-GUESS FIELD DOES NOT EXIST, RAWINS CREATES IT.
 IWIND=1,        ; 1:  USE SURFACE GEOSTROPHIC WIND FOR SFC WIND 1ST GUESS
 ;                    IF USING NMC/OLD ECMWF 1ST GUESS;
 ;                    USE ACTUAL SURFACE WIND IF USING ECTOGA 1ST GUESS
 ;               ; 2:  USE 1000 MB WIND FOR SURFACE WIND FIRST GUESS
 ;
 ;----------DATA INPUT OPTIONS ------------------------------------------
 UNIOBS=.${IfUni}.,       ; TRUE FOR UNIDATA OBS, FALSE FOR ARCHIVED OBS
 ;         RWSUBM=0...ONLY ONE SUBMITTAL INTENDED FOR RAWINS
 ;         RWSUBM=1...FIRST RAWINS SUBMITTAL--TWO ARE INTENDED
 ;         RWSUBM=2...SECOND RAWINS SUBMITTAL
 RWSUBM=${Submit}, ;..SET TO 0,1, OR 2
 IUINTVL=12,       ; TIME INTERVAL (HOURS) FOR RAOB DATA (USUALLY 12).
 ISFCS3 = T,       ; ACQUIRE AND ANALYZE SHIPS AND BUOYS
 ISFCS6 = T,       ; ACQUIRE AND ANALYZE SFC LAND DATA
 F4D = F,          ; CREATE FDDA VOLUME AT NON-STANDARD TIMES.
 INTF4D = 3,       ; TIME INTERVAL FOR FDDA FILE.  IF INTF4D > 3, LAGTEM IS SET
 ;                 ; TO FALSE AUTOMATICALLY IN THE PROGRAM (i.e. USE TEMPORAL
 ;                 ; INTERPOLATION FOR NON-STANDARD TIME FIRST GUESS.)
 LAGTEM = T,       ; TRUE--3 HOUR LAG TIME FOR NON-STD TIME 1ST GUESS
 ;                   FALSE--TEMPORAL INTERPOLATION FOR NON-STD TIME 1ST GUESS
 NSELIM=F,F,F,F,F,F,F,F,F,F,F,         ; DELETE RAOBS-T,F
 NBOGUS=F,F,F,F,F,F,F,F,F,F,F,         ; BOGUS STAT-T,F
 KBOGUS=F,F,F,F,F,F,F,F,F,F,F,         ; BOGUS PTS-T,F
 ;
 ;---------------- DATA QUALITY CONTROL OPTIONS -----------------------
 ;        BUDWGT IS A PARAMETER THAT VARIES THE TOLERANCE OF TH BUDDY
 ;        CHECK. A VALUE OF 1.0 GIVES AN ESTIMATE OF NORMAL TOLERANCE.
 ;        A VALUE > 1.0 INCREASES THE TOLERANCE WHILE A VALUE < 1.0
 ;        DECREASES THE TOLERANCE. A 2.0 WOULD REMOVE ONLY HIGHLY
 ;        SUSPECT DATA WHILE 0.5 WOULD REMOVE SOME GOOD DATA ALONG
 ;        WITH ALL SUSPECT DATA. A VALUE OF 0. OMITS THE BUDDY CHECK.
 ;        IF USING AUTOBOGUS (SUBMIT = 1 OR 2), SET BUDWGT TO ZERO.
 BUDWGT=1.25,         ;    1.25,
 ;
 ;        ERRMX PARAMETERS MAX DIFF ALLOWED BETWEEN FIRST GUESS AND STATION DATA
 ;        IF USING BUDDY CHECK (BUDWGT>0.1), ONLY ONE SUBMITTAL IS EXPECTED.  SET
 ;        LARGER DEFAULT VALUES FOR ERRMX (SUGGESTED VALUES ARE ERRMXT=15,
 ;        ERRMXW=14, AND ERRMXP=8.
 ERRMXT=10.,       ; MAX TEMP DIFF ALLOWED (1ST GUESS MINUS OB)
 ;                 ; 0Z DATA OVER LOW TERRAIN (<500m) FACTORED BY 1.25
```

```
 ;                  ; 0Z DATA OVER HIHG TERRAIN (>=500m) FACTORED BY 1.75
 ;                  ; 12Z DATA FACTORED BY 1.5
 ;                  ; ALL DATA OVER WATER FACTORED BY 0.75
 ERRMXW=13.,        ; MAX WIND DIFF ALLOWED (1ST GUESS MINUS OB)
 ;                  ; DATA LEVEL .LE. 1000.1 IS FACTORED BY 1.25
 ;                  ; DATA LEVEL ABOVE 500 MB IS FACTORED BY 1.5
 ERRMXP=6.,         ; MAX PRES DIFF ALLOWED (1ST GUESS MINUS OB)
 ;                  ; NO LEVEL OR TERRAIN FACTORING
 ;
 ;---------- SET PLOT OPTIONS ----------------------------------------
 ;
 IPLOT=T,F,F,F,T,F,F,F,T,F,F,     ; PLOT VERTICAL RAOBS (As directed
 ;                                ; by ISKEWT).
 ;
 ISKEWT=1,                        ; Active only if IPLOT = T
 ;                                ; 1: Skew-T plots.  2: Stuve Diagrams.
 ;
 ABFLAG=T,T,T,T,T,T,T,T,T,T,T,       ; CONTOUR GRID FIELDS
 ABOVER=T,T,T,T,T,T,T,T,T,T,T,       ; OVERLAY OBSERVATIONS
 ;
 ;---------- PRINT OPTIONS -------------------------------------------
 ISPRNT=F,        ; PRINT SFC INPUT OBS
 IFPRNT=T,        ; PRINT HORIZONTAL FIELDS (FIRST-GUESS AND ANALYSIS)
 & ;-----------------------------------------------------------------
EOF
#
########################################################################
####################                                      #################
####################    END USER MODIFICATION    #################
####################                                      #################
########################################################################
#
if ( $INOBS == UNIOBS ) then
if ( -e unidata.namelist ) rm unidata.namelist
cat > unidata.namelist << EOF
 &UNILIF ;--------NAMELIST INPUT SPECIFICALLY FOR UNIDATA---------------
 UNIUANM = ${#upa_unidate}    ; NUMBER OF UPPER-AIR UNIDATA FILES
 UNISFNM = ${#sfc_unidate}    ; NUMBER OF SURFACE UNIDATA FILES
 & ;-----------------------------------------------------------------
EOF
cat unidata.namelist >> rawins.namelist
endif
#
#    This is where the utility program comes from
#
set MesoUser = /fs/othrorgs/home0/mesouser/Util
#
#
#    The make rules are contained in the file configure.make
#
# --------- create datagrid executable -------------------
#
set LETTERS = (A B C D E F G H I J K L M N O P Q R S T U V W X Y Z)
#
if ( ! -e src/netcdf.incl ) cp /usr/local/include/netcdf.inc src/netcdf.incl
#
echo "using configure.make file with f90 option"
cp configure.make.f90 configure.make
msread src/plots.o /MESOUSER/MM5V2/RAWINS/F90/plots.o
#
```

```
   make
   set toast = $status
   if ( $ENVIRONMENT != BATCH ) then
      echo -n "compile complete, continue?    (yes)    "
      set ans = "$<"
      if (( $ans == "n" ) || ( $ans == "no" )) then
         exit (1)
      endif
   else if (( $ENVIRONMENT == BATCH ) && ( $toast != 0 )) then
      echo "error in the compile, stopping"
      exit(1)
   endif
#
if ( ! -e rawins.exe ) ln -s src/rawins.exe rawins.exe
#
#  edit namelist to remove comments
#
sed -f ./Templates/no_comment.sed rawins.namelist > mif.tmp
mv mif.tmp rawins.namelist
#
# --------- get input files ----------------------------
#
if ( ! -e datagrid ) then
   echo "  msread datagrid $InDatg"
   msread datagrid $InDatg
endif
if ( ! -e datagrid ) then
   echo "  Datagrid file missing.  Could not find MSS file $InDatg"
endif
#
#
if ( $INOBS == UNIOBS ) then
#    acquire netcdf files
#
  set Num = 0
  while ( $Num < ${#upa_unidate})
     @ Num ++
     if ( ! -e data.uao$Num ) then
       if ( -e /wetterhorn/ldm/data/decoded/${upa_unidate[$Num]}_upa.cdf ) then
         cp /wetterhorn/ldm/data/decoded/${upa_unidate[$Num]}_upa.cdf data.uao$Num
       else
         @ MSSDate = ${upa_unidate[$Num]} / 100
         msread data.uao$Num /LDM/${MSSDate}/decoded/${upa_unidate[$Num]}_upa.cdf
       endif
     endif
     if ( ! -e data.uao$Num ) then
        echo " File does not exist; Stopping. "
        exit
     endif
  end
#
  set Num = 0
  while ( $Num < ${#sfc_unidate})
     @ Num ++
     if ( ! -e data.cvt$Num ) then
       if ( -e /wetterhorn/ldm/data/decoded/${sfc_unidate[$Num]}_sao.cdf ) then
       cp /wetterhorn/ldm/data/decoded/${sfc_unidate[$Num]}_sao.cdf data.cvt$Num
       else
         @ MSSDate = ${upa_unidate[$Num]} / 100
         msread data.cvt$Num /LDM/${MSSDate}/decoded/${sfc_unidate[$Num]}_sao.cdf
```

```
       endif
     endif
     if ( ! -e data.cvt$Num ) then
        echo " File does not exist; Stopping. "
        exit
     endif
  end
else
#
#       get RAOB data sets from MS
#
   if ( ! $?InRaobs ) then
      echo "acquiring no RAOB sounding files"
      ja -chlst
      exit (1)
   else if ( ${#InRaobs} >= 1 ) then
      echo "attempting to acquire ${#InRaobs} RAOB sounding file(s)"
   endif
set NUMFIL = 1
while ( $NUMFIL <= ${#InRaobs} )
   set Local  = raobs$LETTERS[${NUMFIL}]
   set Remote = $InRaobs[${NUMFIL}]
      echo "  msread $Local $Remote "
      msread $Local $Remote
   @ NUMFIL ++
end
#
#       get 6-hourly data sets from MS
#
if ( ! $?InSfc6h ) then
   echo "acquiring no 6-hourly surface analysis input files"
   set InSfc6h
else if (( $SFCsw == SFC ) && ( ${#InSfc6h} >= 1 )) then
   echo "attempting to acquire ${#InSfc6h} 6-hourly surface analysis input file(s)"
endif
set NUMFIL = 1
while ( $NUMFIL <= ${#InSfc6h} )
   set Local  = sfc6hr$LETTERS[${NUMFIL}]
   set Remote = $InSfc6h[${NUMFIL}]
      echo "  msread $Local $Remote "
      msread $Local $Remote
   @ NUMFIL ++
end
#
#       get 3-hourly data sets from MS
#
if ( ! $?InSfc3h ) then
   echo "acquiring no 3-hourly surface analysis input files"
   set InSfc3h
else if (( $SFCsw == SFC ) && ( ${#InSfc3h} >= 1 )) then
   echo "attempting to acquire ${#InSfc3h} 3-hourly surface analysis input file(s)"
endif
set NUMFIL = 1
while ( $NUMFIL <= ${#InSfc3h} )
   set Local  = sfc3hr$LETTERS[${NUMFIL}]
   set Remote = $InSfc3h[${NUMFIL}]
      echo "  msread $Local $Remote"
      msread $Local $Remote
   @ NUMFIL ++
end
```

```
endif
#
#       get autobogus file from remote host
#
if (( $Submit == 2 ) && ( ! -e autobog )) then
   rcp ${Host}/AUBG_OUT autobog
   if ( -e autobog ) then
      echo "acquired autobogus file ${Host}/AUBG_OUT from front end machine (via rcp)"
   else
      echo "could not find remote file ${Host}/AUBG_OUT.   Not acquired.  rcp FAILED"
      exit (1)
   endif
endif
#
#       get K bogus file from remote host
#
if ((( $BOGUSsw == Konly ) || ( $BOGUSsw == KandN )) && ( ! -e kbogus )) then
   rcp ${Host}/KBOG_REMOTE kbogus
   if ( -e kbogus ) then
      echo "acquired k bogus file ${Host}/KBOG_REMOTE from front end machine (via rcp)"
   else if ( ! -e kbogus ) then
      echo "could not find remote file ${Host}/KBOG_REMOTE.  Not acquired.  rcp FAILED"
      exit (1)
   endif
endif
#
#       get N bogus file from remote host
#
if ((( $BOGUSsw == Nonly ) || ( $BOGUSsw == KandN )) && ( ! -e nbogus )) then
   rcp ${Host}/NBOG_REMOTE nbogus
   if ( -e nbogus ) then
      echo "acquired n bogus file ${Host}/NBOG_REMOTE from front end machine (via rcp)"
   else if ( ! -e nbogus ) then
      echo "could not find remote file ${Host}/NBOG_REMOTE.  Not acquired.  rcp FAILED"
      exit (1)
   endif
endif
#
#       set up fortran input files for RAWINS
#
if ( -e assign.rawins ) rm assign.rawins
setenv FILENV assign.rawins
   assign -a datagrid             fort.4
   assign -a autobog              fort.10
   assign -a kbogus               fort.12
   assign -a nbogus               fort.13
   assign -a rawins.namelist      fort.14
set NUMFIL = 1
if ( $INOBS != UNIOBS ) then
   while ( $NUMFIL <= 5 )
      @ UNIT = 14 + $NUMFIL
      assign -a raobs$LETTERS[${NUMFIL}] fort.$UNIT
      @ UNIT = 19 + $NUMFIL
      assign -a sfc3hr$LETTERS[${NUMFIL}] fort.$UNIT
      @ UNIT = 24 + $NUMFIL
      assign -a sfc6hr$LETTERS[${NUMFIL}] fort.$UNIT
      if ( -e sfc3hr$LETTERS[${NUMFIL}] ) cp sfc3hr$LETTERS[${NUMFIL}]
shpvol$LETTERS[${NUMFIL}]
      @ UNIT = 29 + $NUMFIL
      assign -a shpvol$LETTERS[${NUMFIL}] fort.$UNIT
```

```
      @ NUMFIL ++
   end
endif
#
#       set up fortran output files for RAWINS
#
assign -a rawanl.out             fort.2
assign -a rawobs.out             fort.11
assign -a raw4dsfc.out           fort.39
assign -a rawab.out              fort.40
assign -a sfc4dobs.out           fort.60
assign -a upr4dobs.out           fort.61
#
# -------- run RAWINS
#
date
if ( -e acct ) rm acct
rawins.exe >&! rawins.print.out
ja -s >! acct
cat acct >> rawins.print.out
if ( $ENVIRONMENT == BATCH ) cat rawins.print.out
#
tar -cf rawins.out.tar src/rawins.exe rawins.namelist rawins.print.out
mswrite -t $RetPd rawins.out.tar ${OutRaw}/rawins_domain${DomId}.out.tar
#
set toast = $status
if (( $ENVIRONMENT == BATCH ) && ( $toast != 0 )) then
   echo "error in the RAWINS, stopping"
   ls -lt
   exit(1)
endif
#
#       if interactive, probably do not want to zap and dispose
#
if ( $ENVIRONMENT != BATCH ) then
   echo -n "test rawins run complete, continue?     (no)    "
   set ans = "$<"
   if (( $ans != "y" ) && ( $ans != "yes" )) then
      rm rawins.namelist assign* core
      exit (1)
   endif
endif
#
#       rawins output files
#
ls -ls
   set test = `ls -l rawanl.out`
   if ( $test[5] < 1000000000 ) then
       mswrite -t $RetPd rawanl.out ${OutRaw}/RAWINS_DOMAIN${DomId}
       echo mswrite -t $RetPd rawanl.out ${OutRaw}/RAWINS_DOMAIN${DomId}

   else
       bsplit -1000000000 rawanl.out  small.raw.
       set NUM = 1
       echo ls small.raw.* >! hold
       set test = `source hold`
       foreach split ( $test )
          mswrite -t $RetPd $split \
                  ${OutRaw}/RAWINS_DOMAIN${DomId}_BSPLIT_$LETTERS[$NUM]
          echo mswrite -t $RetPd $split \
```

```
                  ${OutRaw}/RAWINS_DOMAIN${DomId}_BSPLIT_$LETTERS[$NUM]
          @ NUM ++
       end
       rm small*
   endif

   if ( $OutIEEE == yes ) then
      if ( ! -e ieee.csh ) then
         cp $MesoUser/ieee.csh .
         chmod +x ieee.csh
      endif
      ieee.csh rawanl.out
      set test = `ls -l rawanl.out.ieee`
      if ( $test[5] < 1000000000 ) then
         mswrite -t $RetPd rawanl.out.ieee ${OutRaw}/RAWINS_DOMAIN${DomId}_IEEE
         echo mswrite -t $RetPd rawanl.out.ieee ${OutRaw}/RAWINS_DOMAIN${DomId}_IEEE
      else
         bsplit -1000000000 rawanl.out.ieee small.raw.
         set NUM = 1
         echo ls small.raw.* >! hold
         set test = `source hold`
         foreach split ( $test )
            mswrite -t $RetPd $split \
                  ${OutRaw}/RAWINS_DOMAIN${DomId}_IEEE_BSPLIT_$LETTERS[$NUM]
            echo mswrite -t $RetPd $split \
                  ${OutRaw}/RAWINS_DOMAIN${DomId}_IEEE_BSPLIT_$LETTERS[$NUM]
            @ NUM ++
         end
         rm small*
      endif
   endif

if ( $Submit != 2 ) then
   mswrite -t $RetPd SND.PLT ${OutRaw}/SND.PLT_DOMAIN${DomId}
   rcp SND.PLT ${Host}/SND.PLT
endif
#
if ( -e raw4dsfc.out ) then
   mswrite -t $RetPd raw4dsfc.out ${OutRaw}/RW4DDA_DOMAIN${DomId}
   if ( $OutIEEE == yes ) then
      if ( ! -e ieee.csh ) then
         cp $MesoUser/ieee.csh .
         chmod +x ieee.csh
      endif
      ieee.csh raw4dsfc.out
      mswrite -t $RetPd raw4dsfc.out.ieee ${OutRaw}/RW4DDA_DOMAIN${DomId}_IEEE
   endif
endif
#
mswrite -t $RetPd rawobs.out ${OutRaw}/RAOBS_DOMAIN${DomId}
mswrite -t $RetPd sfc4dobs.out ${OutRaw}/SFC4DOBS_DOMAIN${DomId}
mswrite -t $RetPd upr4dobs.out ${OutRaw}/UPR4DOBS_DOMAIN${DomId}
#
if ( $Submit == 1 ) then
   mswrite -t $RetPd rawab.out ${OutRaw}/AUBG_SUB1_DOMAIN${DomId}
   mswrite -t $RetPd AB.PLT ${OutRaw}/AUB_SUB1_PLT_DOMAIN${DomId}
   rcp rawab.out ${Host}/AUBG_OUT
else if ( $Submit == 2 ) then
   mswrite -t $RetPd autobog ${OutRaw}/AUBG_SUB2_DOMAIN${DomId}
endif
```

```
#
ls -ls
ja -st
```

D.9 interp.deck

```
# QSUB -r INTERP                        # request name
# QSUB -q reg                           # job queue class
# QSUB -eo                              # stdout and stderr together
# QSUB -lM 8Mw                          # maximum memory
# QSUB -lT 1000                         # time limit
# QSUB                                  # no more qsub commands
#
ja
set echo
#
#                ****************************************
#                ******   interp batch C shell   ******
#                ******        V2 System         ******
#                ******   for both cf77 and f90  ******
#                ****************************************
#
if ( $?ENVIRONMENT ) then
   echo "environment variable defined as $ENVIRONMENT"
else
   setenv ENVIRONMENT INTERACTIVE
   echo "environment variable defined as $ENVIRONMENT"
endif
#
if ( $ENVIRONMENT == BATCH ) then
   cd $TMPDIR
endif
#
#       define MSS pathname
#
# this should be the user's case or experiment (for example TEST/EXP1/...).
# Note that there is no / in front of the pathname TEST.
#
   set ExpName = MM5V2/TEST/EXP1        # INTERP MSS input/output pathname
   set RetPd   = 365                    # MSS file retention period in days
#
#       type of interp job
#
   set ForBsw  = FRONT                  # front end job (model input)
#  set ForBsw  = BACK                   # back end job (model output)
#
#  set HYDROsw   = 0                    # hydrostatic INTERP job
   set HYDROsw   = 1                    # nonhydrostatic INTERP job
#
#  set NESTsw  = NoNEST                 # no nest terrain input
   set NESTsw  =   NEST                 # nest terrain input
#
#       locations for input datasets
#
if      ( $ForBsw == FRONT ) then
   set ID     =   1                     # input domain ID
   set InData   =    $ExpName/RAWINS_DOMAIN$ID
else if ( $ForBsw == BACK ) then
   set ID     =   1                     # input domain ID
```

```
   set InData   =   $ExpName/MMOUT_DOMAIN$ID
endif
#
#       if nested, need the fine grid terrain
#
if ( $NESTsw == NEST ) then
   set InTer    =   $ExpName/TERRAIN_DOMAIN2
endif
#
#       INTERP MSS output pathname
#
if ( $HYDROsw == 0 ) then
     set OutInit  =   $ExpName/HY
else
     set OutInit  =   $ExpName/NH
endif
#
#  Do you want to create an additional IEEE copy of the output?
#     IEEE data are used on workstations.
#
   set OutIEEE = no
#  set OutIEEE = yes
#
#  set up the remote host location where your source code may locate
#
   set Host = username@host.domain:/usr/tmp/username
#
#  Do you want to use your source code tar file?
#
   set UseMySource = no
#  set UseMySource = yes
#
#    ------ GET INTERP TAR FILE ---------------------------
#
if ( $UseMySource == yes ) then
     rcp $Host/interp.tar .
     tar -xf interp.tar
else if ( ( $UseMySource == no ) && ( $ENVIRONMENT == BATCH ) ) then
     msread interp.tar.Z /MESOUSER/MM5V2/INTERP/INTERP.TAR.Z
#    cp /fs/othrorgs/home0/mesouser/MM5V2/Interp/interp.tar.Z .
     uncompress interp.tar
     tar -xf interp.tar
endif
rm interp.tar
#
#    ------ INTERP PARAMETER STATEMENT ---------------------------
#
if ( -e src/param.incl ) rm src/param.incl
cat > src/param.incl << EOF
C
C For front end INTERP job, MAXNES and NLNES are the number of domains to be generated
C For back end INTERP standard job, MAXNES and NLNES should alway set to be 1
C For back end INTERP 1WAY NEST job, MAXNES and NLNES should always set to be 2
C
      PARAMETER (MAXNES=2)          ! TOTAL NUMBER OF DOMAINS
      PARAMETER (NLNES=2)           ! NUMBER OF LEVELS OF NEST (SAME AS MM5)
      PARAMETER (IMAX =49, JMAX =52) ! MAX0(IX,IXN1,...)
      PARAMETER (KXS  = 23)         ! VERT COORD: # HALF SIGMA LAYERS
      PARAMETER (KXP  = 21)         ! VERT COORD: # 1000->PTOP
      PARAMETER (KXT  =  1)         ! VERT COORD: # THETA LEVELS (MAXIMUM)
```

```
      PARAMETER (KMAX = 23)          ! MAX0(KXS,KXP,KXT)
EOF
cat >! src/dadlens.incl << EOF
C
C     ... SCALE FACTOR FOR DIRECT ACESS FILES
C
      PARAMETER (IMACHWORD=8) ! Cray
C     PARAMETER (IMACHWORD=4) ! SGI
C
C     ... "SMALL" NUMBER FOR RELAXATION AND PP CONVERGENCE
C
      PARAMETER (SMALLRES=1.0E-9) ! 64 BIT FLOATING POINT
C     PARAMETER (SMALLRES=1.0E-5) ! 32 BIT FLOATING POINT
C
EOF
#    ------ INTERP LOCAL NAMELIST --------------------------------
#           (you need not change anything of the form: ${____})
#     Note: - The first column in the namelist always corresponds to
#             parameters in the first input file
#           - Set STANDARD=.TRUE. if running frontend jobs (except if
#             choosing NOT to remove integrated divergence -
#             see IDIV below), and backend job where you only want to
#             interpolate model output to sigma levels.
#           - Set STANDARD=.FALSE. if it is a one-way backend job or
#             create first-guess from model output data.
#     Referring to inlif_jobtype under /fs/othrorgs/home0/mesouser/MM5V2/Interp
#           to see how to set appropriate namelist for the job.
#
#     (Comments in namelist are removed during execution.)
#
if ( -e inlif ) rm inlif
cat > inlif << EOF
 &LOCMIF ;--------   LOCAL MIF FOR PROGRAM INTERP ---------
 ;
 ;       FULL SIGMA LEVELS FOR FRONT END -- USUALLY REQUIRED
 ;       BOTTOM-UP (SURFACE TO PTOP) ORIENTATION
 NEWCOORD=  1.,.99,.98,.96,.93,.89,.85,.8,.75,.7,.65,.6,
            .55,.5,.45,.4,.35,.3,.25,.2,.15,.1,0.05,0.0, ;
;
 INHYD=${HYDROsw}          ; FRONT END ONLY.  0: HYDROSTATIC,
 ;                          1: NONHYDROSTATIC MODEL INPUT
 STANDARD=.TRUE.,          ; IF TRUE, IGNORE NONSTAN NAMELIST
 ;STANDARD=.FALSE.,        ; IF FALSE, USE NONSTAN NAMELIST ALSO
 BEGTIME = 93031300.,      ; BEGINNING TIME FOR INTERP PROCESSING
 ENDTIME = 93031400.,      ; ENDING TIME FOR INTERP PROCESSING
 ;                          USE 10-DIGIT MDATE FOR V1 DATA AND
 ;                               8-DIGIT MDATE FOR V2 DATA AND ON WORKSTATION
 TIMINT  = 720.,           ; TIME INTERVAL IN MINUTES BETWEEN PROCESS TIME
 NESTIX  = 25, 34, 1,  1,  1,  1,  1,  1,  1,  1,    ; DOMAIN SIZE I
 NESTJX  = 28, 37, 1,  1,  1,  1,  1,  1,  1,  1,    ; DOMAIN SIZE J
 NESTI   = 1, 8, 1, 1,   1,   1,   1,   1,   1,   1, ; START LOCATION I
 NESTJ   = 1, 9,   1,   1,   1,   1,   1,   1,   1,   1, ; START LOCATION J
 LEVIDN  = 0, 1,   2,   0,   0,   0,   0,   0,   0,   0,
 ;                                    LEVEL OF NEST FOR EACH DOMAIN
 NUMNC   = 1,   1,   2,   1,   1,   1,   1,   1,   1,   1,
 ;                                    ID OF MOTHER DOMAIN FOR EACH NEST
 OLDTER  =.FALSE.                     ; IF FALSE, USE NESTED TERRAIN FILE
 ;OLDTER  =.TRUE.                     ; IF FALSE, USE NESTED TERRAIN FILE
 IRATIO  = 3,                         ; NEST RATIO
 & ;--------------------------------------------------------------
```

```
 &NHYDMIF  ;----   ONLY REQUIRED FOR FRONT-END NONHYDROSTATIC INTERP JOB --
 TS0      = 275.,           ; SEA-LEVEL TEMPERATURE OF REF STATE (K)
 TLP      = 50.,            ; LAPSE RATE d(T)/d(ln P) OF REF STATE (K)
 P0       = 1.E+05          ; SEA-LEVEL PRESSURE (Pa)
 & ;---------------------------------------------------------------------
 &NONSTAN  ;----   ONLY REQUIRED FOR NON-STANDARD INTERP JOBS   ----
 ;          use only if IUSE=1
 ICUT1    = 1,              ; I=1 LOCATION OF SUBDOMAIN IN BIG GRID
 ICUT2    = 25,             ; I=IMAX LOCATION OF SUBDOMAIN IN BIG GRID
 JCUT1    = 1,              ; J=1 LOCATION OF SUBDOMAIN IN BIG GRID
 JCUT2    = 28,             ; J=JMAX LOCATION OF SUBDOMAIN IN BIG GRID
 ;
 ;           *** REPLACE DEFAULTS WITH THESE VALUES ***
 IUSE     = 1,              ; 0 = INTERP USES NEWCOORD + REPLACE DEFAULTS
                            ; 1 = OUTPUT DATA IN SPECIAL FORMAT
 OUTPUTONCE=.FALSE.,      ; TRUE  = OUTPUT INITIAL CONDITION TIME PERIOD ONLY
                            ;     WHEN GENERATING 1-WAY MMINPUT FILE
                            ; FALSE = OUTPUT ALL TIMES
 MMRESTEMP=.FALSE.,         ; TRUE  = USE COARSE DOMAIN SUBSTRATE TEMP
                            ; FALSE = RECALCULATE
 MAKEDATA= 2,               ; IF IUSE=1, THEN EXTRA DATA TO DISPOSE
                            ;     1 = MM AS FIRST GUESS TO RAWINS
                            ;     2 = 1 WAY NEST
                            ;     3 = PRESSURE LEVEL INPUT FOR VERIFY
 IDIV     = 0,              ; FRONT END ONLY
                            ;     0 = REMOVE INTEGRATED MEAN DIVERGENCE
                            ;     1 = DO NOT REMOVE INTEGRATED MEAN DIVERGENCE
 & ;---------------------------------------------------------------------
EOF
##############################################################################
####################                                 #######################
####################    END USER MODIFICATION        #####################
####################                                 #######################
##############################################################################
#
#    This is where the utility program comes from
#
set MesoUser = /fs/othrorgs/home0/mesouser/Util
#
#    The make rules are contained in the file configure.make
#
# --------- create interp executable ------------------
#
set LETTERS = (A B C D E F G H I J K L M N O P Q R S T U V W X Y Z)
set DomId = (1 2 3 4 5 6 7 8 9 10)
#
echo "using configure.make file with f90 option"
cp Templates/configure.make.f90 configure.make
#
   make
   set toast = $status
   if ( $ENVIRONMENT != BATCH ) then
      echo -n "compile complete, continue?    (yes)   "
      set ans = "$<"
      if (( $ans == "n" ) || ( $ans == "no" )) then
         exit (1)
      endif
   else if (( $ENVIRONMENT == BATCH ) && ( $toast != 0 )) then
      echo "error in the compile, stopping"
      exit(1)
```

```
   endif
#
if ( ! -e interp.exe ) ln -s src/interp.exe interp.exe
#
#  edit namelist to remove comments
#
sed -f ./Templates/no_comment.sed inlif > mif.tmp
mv mif.tmp inlif
#
# --------- get data sets from MS
#
if        ( $ForBsw == FRONT ) then
   echo "attempting to acquire ${#InData} front end analysis file"
else if (( $ForBsw == BACK ) && ( ${#InData} == 1 )) then
   echo "attempting to acquire ${#InData} back end model output file"
else if (( $ForBsw == BACK ) && ( ${#InData} > 1 )) then
   echo "attempting to acquire ${#InData} back end model output files"
endif
set NUMFIL = 1
while ( $NUMFIL <= ${#InData} )
   set Local  = interpin$LETTERS[${NUMFIL}]
   set Remote = $InData[${NUMFIL}]
   msread $Local $Remote
   @ NUMFIL ++
end
#
# -------- get fine grid terrain file from MS
#
if ( $NESTsw == NEST ) then
set NUMFIL = 1
while ( $NUMFIL <= ${#InTer} )
   set Local  = terrain$LETTERS[${NUMFIL}]
   set Remote = $InTer[${NUMFIL}]
   msread $Local $Remote
   @ NUMFIL ++
end
endif
#
# -------- set up fortran input files for INTERP
#
set UNIT   = 0
set NUMFIL = 1
if ( -e assign.interp ) rm assign.interp
setenv FILENV assign.interp
while ( $NUMFIL <= ${#InData} )
   set Local = interpin$LETTERS[${NUMFIL}]
   @ UNIT = 49 + $NUMFIL
   assign -a $Local             fort.$UNIT
   @ NUMFIL ++
end
#
# -------- set up fortran input files for NEST TERRAIN FILES
#
if ( $NESTsw == NEST ) then
set UNIT   = 0
set NUMFIL = 1
while ( $NUMFIL <= ${#InTer} )
   set Local = terrain$LETTERS[${NUMFIL}]
   @ UNIT = 39 + $NUMFIL
   assign -a $Local             fort.$UNIT
```

```
   @ NUMFIL ++
end
endif
assign -a inlif                 fort.36
#
# -------- set up fortran output files for INTERP
#
assign -a bndtgd  fort.30
assign -a firstg  fort.34
#
#       set up fortran temporary files for INTERP
#
assign -a bndry                 fort.31
assign -a bnd2d                 fort.32
assign -a bnd3d                 fort.33
assign -a int2d                 fort.90
assign -a int3d                 fort.91
#
# --------- run INTERP
#
date
interp.exe >&! interp.print.out
set toast = $status
if (( $ENVIRONMENT == BATCH ) && ( $toast != 0 )) then
   cat interp.print.out
   echo "error in interp, stopping"
   rm core
   exit(1)
endif
#
#       if interactive, probably do not want to zap and dispose
#
if ( $ENVIRONMENT != BATCH ) then
   echo -n "test interp run complete, continue?     (no)    "
   set ans = "$<"
   if (( $ans != "y" ) && ( $ans != "yes" )) then
      rm inlif assign* fort.9* core
      exit (1)
   endif
endif
#
if ( -e acct ) rm acct
ja -s >! acct
cat acct >> interp.print.out
if ( $ENVIRONMENT == BATCH ) cat interp.print.out
#
# -------- output files
#
ls -ls
if (( $ForBsw == FRONT )) then
#
#               mm5 initial and nest rawins files
#
set Tens   = (   2       6    )
set Name   = ( MMINPUT RAWINS )
foreach OutType ( 1 2 )
   set OutFileType = $Name[$OutType]
   foreach Units ( 1 2 3 4 5 6 7 8 9 )
      if ( -e fort.$Tens[$OutType]${Units} ) then
         echo ls -l fort.$Tens[$OutType]${Units} >! hold
```

```
         set test = `source hold`
         if ( $test[5] < 1000000000 ) then
            mswrite -t $RetPd fort.$Tens[$OutType]$Units \
                    $OutInit/${OutFileType}_DOMAIN${Units}
         else
            bsplit -1000000000 fort.$Tens[$OutType]${Units} \
                   small.$Tens[$OutType]${Units}.
            set NUM = 1
            echo ls small.$Tens[$OutType]${Units}.* >! hold
            set test = `source hold`
            foreach split ( $test )
               mswrite -t $RetPd $split \
                  $OutInit/${OutFileType}_DOMAIN${Units}_BSPLIT_$LETTERS[$NUM]
               @ NUM ++
            end
            rm small.*
         endif
      endif
   end
end
#
#                  bdyout file
#
if ( -e bndtgd ) then
   set test = `ls -l bndtgd`
   if ( $test[5] < 1000000000 ) then
      mswrite -t $RetPd bndtgd $OutInit/BDYOUT_DOMAIN${ID}
   else
      bsplit -1000000000 bndtgd small.bndtgd.
      set NUM = 1
      echo ls small.bndtgd.* >! hold
      set test = `source hold`
      foreach split ( $test )
         mswrite -t $RetPd $split \
            $OutInit/BDYOUT_DOMAIN${ID}_BSPLIT_$LETTERS[$NUM]
         @ NUM ++
      end
      rm small.*
   endif
endif
#                 verify input files interpolated from rawins
#
set Numb = ${ID}
set IfAny = ( `ls fort.8*` )
set MAX = ${#IfAny}
if ( $MAX > 0 ) then
   foreach fil ( `ls fort.8*` )
   echo "mswrite -t $RetPd $fil $OutInit/VERIFY_RAWINS_DOMAIN${DomId[$Numb]}"
   mswrite -t $RetPd $fil $OutInit/VERIFY_RAWINS_DOMAIN${DomId[$Numb]}
   @ Numb ++
   end
endif
if ( $OutIEEE == yes ) then
     if ( ! -e ieee.csh ) then
         cp $MesoUser/ieee.csh .
         chmod +x ieee.csh
     endif
     set NUMFILE = 1
     foreach file ( fort.2[1-9] )
         echo $file
```

```
         if ( $file == fort.21 ) then
            ieee.csh $file bndtgd
            set test = `ls -l bndtgd`
            if ( $test[5] < 1000000000 ) then
               mswrite -t $RetPd bndtgd.ieee $OutInit/BDYOUT_DOMAIN${ID}_IEEE
            else
               bsplit -1000000000 bndtgd.ieee small.bndtgd.
               set NUM = 1
               echo ls small.bndtgd.* >! hold
               set test = `source hold`
               foreach split ( $test )
                  mswrite -t $RetPd $split \
                     $OutInit/BDYOUT_DOMAIN${ID}_IEEE_BSPLIT_$LETTERS[$NUM]
                  @ NUM ++
               end
            endif
         else
            ieee.csh $file
         endif
         set test = `ls -l $file.ieee`
         if ( $test[5] < 1000000000 ) then
            mswrite -t $RetPd $file.ieee $OutInit/MMINPUT_DOMAIN${NUMFILE}_IEEE
         else
            bsplit -1000000000 $file.ieee small.$file.
            set NUM = 1
            echo ls small.$file.* >! hold
            set test = `source hold`
            foreach split ( $test )
               mswrite -t $RetPd $split \
                  $OutInit/MMINPUT_DOMAIN${NUMFILE}_IEEE_BSPLIT_$LETTERS[$NUM]
               @ NUM ++
            end
          endif
         @ NUMFILE ++
     end
endif
else
#                  interpolated mm5 output and/or 1-way output files
#
set Tens   = (   2         6       )
set Name   = ( MMOUTP MMINPUT_1WAY )
foreach OutType ( 1 2 )
   set OutFileType = $Name[$OutType]
   foreach Units ( 1 2 3 4 5 6 7 8 9 )
      if ( -e fort.$Tens[$OutType]${Units} ) then
         echo ls -l fort.$Tens[$OutType]${Units} >! hold
         set test = `source hold`
         if ( $test[5] < 1000000000 ) then
            mswrite -t $RetPd fort.$Tens[$OutType]$Units \
                    $OutInit/${OutFileType}_DOMAIN${Units}
         else
            bsplit -1000000000 fort.$Tens[$OutType]${Units} \
                   small.$Tens[$OutType]${Units}.
            set NUM = 1
            echo ls small.$Tens[$OutType]${Units}.* >! hold
            set test = `source hold`
            foreach split ( $test )
               mswrite -t $RetPd $split \
                  $OutInit/${OutFileType}_DOMAIN${Units}_BSPLIT_$LETTERS[$NUM]
               @ NUM ++
```

```
            end
            rm small.*
         endif
      endif
   end
end
#                  bdyout file for 1-way run
#
if ( -e bndtgd ) then
   set ID2 = 0
   @ ID2 = $ID + 1
   set test = `ls -l bndtgd`
   if ( $test[5] < 1000000000 ) then
      mswrite -t $RetPd bndtgd $OutInit/BDYOUT_1WAY_DOMAIN${ID2}
   else
      bsplit -1000000000 bndtgd small.bndtgd.
      set NUM = 1
      echo ls small.bndtgd.* >! hold
      set test = `source hold`
      foreach split ( $test )
         mswrite -t $RetPd $split \
            $OutInit/BDYOUT_1WAY_DOMAIN${ID2}_BSPLIT_$LETTERS[$NUM]
         @ NUM ++
      end
      rm small.*
   endif
endif
#                  first guess back to rawins
#
if ( -e firstg ) then
   set test = `ls -l firstg`
   if ( $test[5] < 1000000000 ) then
      mswrite -t $RetPd firstg $OutInit/FIRSTG_DOMAIN${ID}
   else
      bsplit -1000000000 firstg small.firstg.
      set NUM = 1
      echo ls small.firstg.* >! hold
      set test = `source hold`
      foreach split ( $test )
         mswrite -t $RetPd $split \
            $OutInit/FIRSTG_DOMAIN${ID}_BSPLIT_$LETTERS[$NUM]
         @ NUM ++
      end
      rm small.*
   endif
endif
#                   verify input files from interpolated mm5 output
#
if ( -e fort.80 ) then
   echo "mswrite -t $RetPd fort.80 $OutInit/VERIFY_MMOUT_DOMAIN${ID}"
   mswrite -t $RetPd fort.80 $OutInit/VERIFY_MMOUT_DOMAIN${ID}
endif
endif
#
# tar the namelist, executable, and output together, save the INTERP tar file on MSS
#
tar -cf interp.out.tar inlif interp.print.out src/interp.exe
mswrite -t $RetPd interp.out.tar $OutInit/interp.out.tar
ja -st
```

D.10 For Cray batch jobs:

```
# QSUB -r MM5v2                          # request name
# QSUB -q econ                           # job queue class
# QSUB -eo                               # stdout and stderr together
# QSUB -lM 10Mw                          # maximum memory
# QSUB -lT 9000                          # time limit
# QSUB                                   # no more qsub commands
#
ja
set echo
#
#
#               ******************************************
#               ******* mm5 interactive/batch C shell ******
#               *******          V2 System           ******
#               *******     for both cf77 and f90     ******
#               ******************************************
#
Note: when running this deck interactively, it assumes
#           the source files are local and un-tared.
#
#     CAUTION: If you have changed/added new options in configure.user,
#              you have to re-compile the code, i.e. set compile = yes
#
#
if ( $?ENVIRONMENT ) then
   echo "environment variable defined as $ENVIRONMENT"
else
   setenv ENVIRONMENT INTERACTIVE
   echo "environment variable defined as $ENVIRONMENT"
endif
#
if ( $ENVIRONMENT == BATCH ) then
   cd $TMPDIR
endif
#
#    how many CRAY CPUs to use to run the model, set to 1 if not multitasking
#
setenv NCPUS 8
#
#       this should be the user's case or experiment (used in MS name)
#
  set ExpName    = MM5/TEST           # MSS path name for output
  set InName     = MM5/TEST           # MSS path name for input
  set RetPd      = 365                # MSS retention period in days
#
#       type of mm5 job
#
  set compile    = yes                # if yes, compile the mm5 code
# set compile    = no                 # if no, the deck expects to read mm5.exe

  set execute    = yes                # if yes, execute model run
# set execute    = no                 # if no, complie only
#
  set UseMySource = no                # Use "standard" version of MM5
# set UseMySource = yes               # Use my version of MM5
#
  set CaseName   = CTL                # MSS pathname for this experiment
#
  set STARTsw    = NoReStart          # start model run at hour 0
# set STARTsw    = ReStart            # restart model run
#
  if ( $STARTsw == ReStart ) then
     set compile = no
  endif
#
  set FDDAsw     = NoFDDA             # no FDDA input files
# set FDDAsw     = Anly               # gridded FDDA input files
# set FDDAsw     = Obs                # obs FDDA input files
# set FDDAsw     = Both               # gridded and obs FDDA input files
#
```

```
# set HYDROsw    = Hydro               # hydrostatic input files
  set HYDROsw    = NonHydro            # nonhydrostatic input files
#
  if ( $HYDROsw == NonHydro ) then
     set InitName  =  $InName/NH
  else
     set InitName  =  $InName/HY
  endif
#
#    Input files:
#        1. boundary file
#
  set InBdy  =  $InitName/BDYOUT_DOMAIN1
#
#        2. initial conditions
#
# set InMM     = ( $InitName/MMINPUT_DOMAIN1 )
  set InMM     = ( $InitName/MMINPUT_DOMAIN1 \
                   $InitName/MMINPUT_DOMAIN2 )
#
#        3. input restart file - if this is a restart run
#
  if ( $STARTsw == ReStart ) then
     set InRst     = ( $ExpName/${CaseName}/SAVE_DOMAIN1 \
                       $ExpName/${CaseName}/SAVE_DOMAIN2 )
  endif
#
#        4. 4dda surface analyses - if this is a 4DDA run
#       provide fake name(s) if not doing surface 4DDA
#
if (( $FDDAsw == Anly ) || ( $FDDAsw == Both )) then
  set In4DSfc  = ( $InName/RW4DDA_DOMAIN1 \
                   $InName/RW4DDA_DOMAIN2  )
endif
#
#        5. 4dda observations - if this is a 4DDA run
#
if (( $FDDAsw == Obs  ) || ( $FDDAsw == Both )) then
  set In4DObs  = ( $InName/MM5OBS_DOMAIN1 \
                   $InName/MM5OBS_DOMAIN2 )
endif
#
#    Output files: MSS path name for output, save,
#                  shut down and print out files
#
if ( $STARTsw == ReStart ) then
   set OutMM       = ${ExpName}/${CaseName}_RES
else
   set OutMM       = ${ExpName}/$CaseName
endif
#
#      set up the remote host location where your source code may locate
#
  set Host     = username@host.domain:/usr/tmp/username
#
#
#============================================================
#
#       Get source code
#
if ( $ENVIRONMENT == BATCH ) then
   if ( $compile == yes ) then
      if ( $UseMySource == yes ) then
         rcp $Host/mm5v2.tar .
#        msread mm5v2.tar MM5/TEST/MM5V2.TAR
      else
         msread mm5v2.tar /MESOUSER/MM5V2/MM5/MM5V2.TAR
#        cp /fs/othrorgs/home0/mesouser/MM5V2/MM5/mm5v2.tar.Z .
#        uncompress mm5v2.tar
      endif
   else
      msread mm5v2.tar ${ExpName}/${CaseName}/mm5exe.tar
   endif
   tar -xf mm5v2.tar
```

```
   rm mm5v2.tar
   ls -ls
endif
#
#=============================================================
# configure.user file - User editting required for sections 5 and 6
#  **Please read Release-note for how to make changes in configure.user
#-----------------------------------------------------------------------
#
cat >! ./configure.user << "EOF"
# Sections
# 1. System Variables
# 3. Fortran options
#    3a. Cray (YMP, J90)
# 4. General commands
# 5. Options for making "./include/parame.incl"
# 6. Physics Options (memory related)
# 7. make rules - Don't need to change
#
#-----------------------------------------------------------------------
# 1. System Variables
#-----------------------------------------------------------------------
SHELL =         /bin/sh
.SUFFIXES: .F .i .o .f
#----------------------------------------------------------------------------
# 2. User Variables
#----------------------------------------------------------------------------
# RUNTIME_SYSTEM: do not change if using NCAR's Cray
#
RUNTIME_SYSTEM = "CRAY"
#-----------------------------------------------------------------------
# 3. Fortran options
#-----------------------------------------------------------------------
LIBINCLUDE = $(DEVTOP)/include
#-----------------------------------------------------------------------
#    3a. Cray
#    Note: - imsl library is only needed if running Arakawa-Schubert cumulus scheme;
#          and the location of the library may be different on non-NCAR Crays.
#          - if you are using the new program environment on Cray, should set
#          CPP = /opt/ctl/bin/cpp
#          - select the right compilation option for Cray - you may use
#          f90 option on paiute
#
#-----------------------------------------------------------------------
FC = f90
FCFLAGS = -D$(RUNTIME_SYSTEM) -I$(LIBINCLUDE) -O task1
CFLAGS =
CPP = /opt/ctl/bin/cpp
CPPFLAGS = -I$(LIBINCLUDE) -C -P
LDOPTIONS =
LOCAL_LIBRARIES = -L /usr/local/lib -l imsl
MAKE = make -i -r
#-----------------------------------------------------------------------
# 4. General commands
#-----------------------------------------------------------------------
AR = ar ru
RM = rm -f
RM_CMD = $(RM) *.CKP *.ln *.BAK *.bak *.o *.i core errs ,* *~ *.a \
.emacs_* tags TAGS make.log MakeOut *.f !
GREP = grep -s
CC = cc
#-----------------------------------------------------------------------
# 5. Options for making ./include/parame.incl
#-----------------------------------------------------------------------
#
# NHYDRO (integer)                    - "1" -> NonHydrostatic run
#                                     - "0" -> Hydrostatic run
NHYDRO = 1
# FDDAGD (integer)                    - "1" -> FDDA gridded run
#                                     - "0" -> NonFDDA run
FDDAGD = 0
# FDDAOBS (integer)                   - "1" -> FDDA obs run
#                                     - "0" -> NonFDDA run
FDDAOBS = 0
```

```
# MAXNES (integer)                   - Max Number of Domains in simulation
MAXNES = 2
# MIX,MJX,MKX (integer)              - Maximum Dimensions of any Domain
MIX = 49
MJX = 52
MKX = 23
#-----------------------------------------------------------------------
# 6. Physics Options
#    The first MAXNES values in the list will be used for the corresponding
#        model nests; the rest in the list can be used to compile other options.
#        The exception is FRAD, of which only the first value is used in the model,
#        (i.e., only one radiation option is used for all nests). The rest allow
#        other options to be compiled.
#-----------------------------------------------------------------------
# IMPHYS - for explicit moisture schemes (array,integer)
IMPHYS = "4,4,1,1,1,1,1,1,1,1"
#                                    - Dry,stable precip.,warm rain,simple ice
#                                    - 1  ,2             ,3        ,4
#                                    - mix phase,graupel(gsfc),graupel(reisner2)
#                                    - 5        ,6            ,7
MPHYSTBL = 0
#                                    - 0=do not use look-up tables for moist
#                                      physics
#                                    - 1=use look-up tables for moist physics
#                                      (currently only simple ice and mix phase
#                                       are available)
#
# ICUPA - for cumulus schemes (array,integer)
#                                    - None,Kuo,Grell,AS,FC,KF,BM - 1,2,3,4,5,6,7
ICUPA  = "3,3,1,1,1,1,1,1,1,1"
#
# IBLTYP - for planetary boundary layer (array,integer)
#                                    - PBL type 0=no PBL fluxes,1=bulk,
#                                      2=Blackadar,3=Burk-Thompson,5=MRF(ISOIL=1)
IBLTYP  = "5,5,1,1,1,1,1,1,1,1"
#
# FRAD - for atmospheric radiation (integer)
#                                    - Radiation cooling of atmosphere
#                                      0=none,1=simple,2=cloud,3=ccm2
FRAD = "2,0,0,0,0"
#
# ISOIL - for multi-layer soil temperature model (integer)
#                                    - 0=no,1=yes (only works with IBLTYP=2,5)
ISOIL = 1
#
# ISHALLO (array,integer)            - Shallow Convection Option
#                                      1=shallow convection,0=No shallow convection
ISHALLO  = "0,0,0,0,0,0,0,0,0,0"
#-----------------------------------------------------------------------
# 7. make rules - Don't need to change
#-----------------------------------------------------------------------
.F.i:
	$(RM) $@
	$(CPP) $(CPPFLAGS) $*.F > $@
	mv $*.i $(DEVTOP)/pick/$*.f
	cp $*.F $(DEVTOP)/pick
.c.o:
	$(RM) $@ && \
	$(CC) -c $(CFLAGS) $*.c
.F.o:
	$(RM) $@
	$(FC) -c $(FCFLAGS) $*.F
.F.f:
	$(RM) $@
	$(CPP) $(CPPFLAGS) $*.F > $@
.f.o:
	$(RM) $@
	$(FC) -c $(FCFLAGS) $*.f
#----------------------------------------------------------------------
"EOF"
#
#=========================================================
# Options for namelist ("mmlif") - User editting required
#   (Comments in the namelist are removed during execution)
```

```
#-----------------------------------------------------------------------
#
# Note: The first dimension (column) of the arrays denotes the domain
#       identifier.
#          Col 1 = Domain #1, Col 2 = Dom #2, etc.
#       If there is only one colume, the choice is true for all domains.
#
cat >! ./Run/oparam << EOF
 &OPARAM
 ;
 ;      ************** OUTPUT/RESTART OPTIONS ***************
 ;
 IFREST = .FALSE.,                ; whether this is a restart
    IXTIMR  = 720,                ; restart time in minutes
 LEVIDN = 0,1,1,1,1,1,1,1,1,1,    ; level of nest for each domain
 NUMNC = 1,1,1,1,1,1,1,1,1,1,     ; ID of mother domain for each nest
 IFSAVE = .TRUE.,                 ; save data for restart
    SVLAST = .FALSE.,             ; T: only save the last file for restart
                                  ; F: save multiple files
    SAVFRQ =360.,                 ; how frequently to save data (in minutes)
 IFTAPE = 1,                      ; model output: 0,1
    TAPFRQ =60.,         ; how frequently to output model results (in minutes)
    INCTAP = 1,1,1,1,1,1,1,1,1,1, ; multipliers of TAPFRQ for outputting
 IFSKIP = .FALSE.,       ; whether to skip input files - DO NOT use this for restart
    MDATEST = 00000000,           ; the MDATE for the start file
 IFPRT = 0,                       ; sample print out: =1, a lot of print
 PRTFRQ = 720.,                   ; Print frequency for sample output (in minutes)
 MASCHK = 60,            ; mass conservation check (KTAU or no. of time steps)
 &END
EOF
cat >! ./Run/lparam << EOF
 &LPARAM
 ;
 ;       1. user-chosen options I
 ;
 RADFRQ = 30.,       ;atmospheric radiation calculation frequency (in minutes)
 IMVDIF = 1,         ;moist vertical diffusion in clouds - 0, 1 (IBLTYP=2,5 only)
 IVQADV = 1,         ;vertical moisture advection uses log interpolation - 0,
                     ;linear - 1
 IVTADV = 1,         ;vertical temperature advection uses theta interpolation- 0,
                     ;linear - 1
ITHADV = 1,          ;advection of temperature uses potential temperature - 1,
                     ;standard - 0
 ITPDIF = 1,         ;diffusion using perturbation temperature- 0,1 (NH run only)
 ICOR3D = 1,         ;3D Coriolis force (for NH run only) - 0, 1
 IFUPR = 1,               ;upper radiative boundary condition (NH) - 0, 1
 ;
 ;       2. keep the following two variables as they are
 ;          unless doing sensitivity runs
 ;
 IFDRY     = 0,           ;fake-dry run (no latent heating) - 0, 1
 ;                        for IMPHYS = 2,3,4,5,6,7 (requires ICUPA = 1)
 ICUSTB    = 1,           ;stability check for Anthes-Kuo CPS only - 0, 1
 ;
 ;       3. user-chosen options II
 ;
 IBOUDY = 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, ;boundary conditions
 ;            (fixed, relaxation, time-dependent,
 ;            time and inflow/outflow dependent (HY)/relaxation (NH),
 ;            SPONGE (HY only) - 0, 1, 2, 3, 4)
 IFSNOW = 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ;SNOW COVER EFFECTS - 0, 1
 ;            (only if snow data are generated in DATAGRID)
 ;
 ;       4. keep the following 9 variables as they are
 ;          unless doing sensitivity runs
 ;
 ISFFLX = 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ;surface fluxes - 0, 1
 ITGFLG = 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ;surface temperature prediction - 1, 3
 ISFPAR = 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ;surface characteristics - 0, 1
 ICLOUD = 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ;cloud effects on radiation - 0, 1
 ;                                       currently for IFRAD = 1,2
 ICDCON = 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ;constant drag coefficients - 0, 1
 ;                                       (IBLTYP=1 only)
 IVMIXM = 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ;vertical mixing of momentum - 0, 1
```

```
 ;                                         (IBLTYP=2 only)
 HYDPRE = 1.,1.,1.,1.,1.,1.,1.,1.,1.,1.,;HYDRO EFFECTS OF LIQ WATER - 0., 1.
 ;                                         (HY run only)
 IEVAP  = 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ;EVAP OF CLOUD/RAINWATER - <0, 0, >0
 ;                                         (currently for IMPHYS=3,4,5 only)
 ;
 ;
 ;       ************* NESTING OPTIONS ****************
 ;
 NESTIX =  35, 49,  1,  1,  1,  1,  1,  1,  1,  1,    ; domain size i
 NESTJX =  41, 52,  1,  1,  1,  1,  1,  1,  1,  1,    ; domain size j
 NESTI  =   1, 10,  1,   1,   1,   1,   1,   1,   1,   1,   ; start location i
 NESTJ  =   1, 17,  1,   1,   1,   1,   1,   1,   1,   1,   ; start location i
 XSTNES =   0.,  0., 0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,
                                                      ; domain initiation
 XENNES =1440.,1440.,720.,720.,720.,720.,720.,720.,720.,720.,
                                                      ; domain termination
 IOVERW =   1,   1,   0,   0,   0,   0,   0,   0,   0,   0,
                                                      ; overwrite nest input
 ;          0=interpolate from coarse mesh (for nest domains);
 ;          1=read in domain initial conditions
  ;          2=interpolate from coarse mesh, but read in high-resolution
  ;          terrain and landuse
 IACTIV =   1,   0,   0,   0,   0,   0,   0,   0,   0,   0,  ;
 ;          in case of restart: is this domain active?
 ;
 ;
 ;       ************* MOVING NEST OPTIONS ******************
 ;
 IMOVE  =   0,   0,   0,   0,   0,   0,   0,   0,   0,   0, ; move domain 0,1
 IMOVCO =    1,   1,   1,   1,   1,   1,   1,   1,   1,   1, ; 1st move #
 ;          imovei(j,k)=L, ; I-INCREMENT MOVE (DOMAIN J, MOVE NUMBER K) IS L
 IMOVEI =    0,   0,   0,   0,   0,   0,   0,   0,   0,   0, ; I move #1
            0,   0,   0,   0,   0,   0,   0,   0,   0,   0, ; I move #2
            0,   0,   0,   0,   0,   0,   0,   0,   0,   0, ; I move #3
            0,   0,   0,   0,   0,   0,   0,   0,   0,   0, ; I move #4
            0,   0,   0,   0,   0,   0,   0,   0,   0,   0, ; I move #5
 IMOVEJ =    0,   0,   0,   0,   0,   0,   0,   0,   0,   0, ; J move #1
            0,   0,   0,   0,   0,   0,   0,   0,   0,   0, ; J move #2
            0,   0,   0,   0,   0,   0,   0,   0,   0,   0, ; J move #3
            0,   0,   0,   0,   0,   0,   0,   0,   0,   0, ; J move #4
            0,   0,   0,   0,   0,   0,   0,   0,   0,   0, ; J move #5
 IMOVET =    0,   0,   0,   0,   0,   0,   0,   0,   0,   0, ; time of move #1
            0,   0,   0,   0,   0,   0,   0,   0,   0,   0, ; time of move #2
            0,   0,   0,   0,   0,   0,   0,   0,   0,   0, ; time of move #3
            0,   0,   0,   0,   0,   0,   0,   0,   0,   0, ; time of move #4
            0,   0,   0,   0,   0,   0,   0,   0,   0,   0, ; time of move #5
EOF
cat >! ./Run/pparam << EOF
 &PPARAM
 ;
 ;       ************* FORECAST TIME AND TIME STEP ******************
 ;
 TIMAX  = 720.,          ; forecast length in minutes
 TISTEP = 270.,           ; coarse domain DT in model, use 3*DX
 ;
 ;       ************* MISCELLANEOUS OPTIONS *****************
 ;
 ; The values for the following 5 variables are only used if ISFPAR = 0
 ;      (i.e. only land/water surface catagories)
 ;
 ZZLND  = 0.1,            ; roughness length over land in meters
 ZZWTR  = 0.0001,         ; roughness length over water in meters
 ALBLND = 0.15,           ; albedo
 THINLD = 0.04,           ; surface thermal inertia
 XMAVA  = 0.3,               ; moisture availability over land as a decimal
                             ; fraction of one
  CONF   = 1.0,              ; non-convective precipitation saturation threshold
                             ;(=1: 100%)
 IFEED  = 3,              ; no feedback, 9-pt weighted average, 1-pt feedcak with
                             ;no smoothing / light smoothing / heavy smoothing
                          ; - 0,1,2,3, and 4
 IABSOR = 0,              ; sponge upper boundary (HYD only) - 0,1
 &END
```

```
EOF
cat >! ./Run/fparam << EOF
 &FPARAM
 ;
 ;       ************* 4DDA OPTIONS **********************
 ;
 ;      THE FIRST DIMENSION (COLUMN) IS THE DOMAIN IDENTIFIER:
 ;          COLUMN 1 = DOMAIN #1, COLUMN 2 = DOMAIN #2, ETC.
 ;
 ;        START TIME FOR FDDA (ANALYSIS OR OBS) FOR EACH DOMAIN
 ;        (IN MINUTES RELATIVE TO MODEL INITIAL TIME)
 FDASTA=0.,0.,0.,0.,0.,0.,0.,0.,0.,0.
 ;        ENDING TIME FOR FDDA (ANALYSIS OR OBS) FOR EACH DOMAIN
 ;        (IN MINUTES RELATIVE TO MODEL INITIAL TIME)
 FDAEND=780.,0.,0.,0.,0.,0.,0.,0.,0.,0.,
 ;
 ;       **************** ANALYSIS NUDGING ******************
 ;
 ;      THE FIRST DIMENSION (COLUMN) OF THE ARRAYS DENOTES THE
 ;      DOMAIN IDENTIFIER:
 ;          COLUMN 1 = DOMAIN #1, COLUMN 2 = DOMAIN #2, ETC.
 ;      THE SECOND DIMENSION (ROW OR LINE) EITHER REFERS TO THE 3D VS
 ;      SFC ANALYSIS OR WHICH VARIABLE IS ACCESSED:
 ;          LINE 1 = 3D, LINE 2 = SFC   OR
 ;          LINE 1 = U, LINE 2 = V, LINE 3 = T, LINE 4 = Q
 ;
 ;        IS THIS A GRID 4DDA RUN? 0 = NO; 1 = YES
 I4D= 0,0,0,0,0,0,0,0,0,0,
      0,0,0,0,0,0,0,0,0,0,
 ;
 ;        SPECIFY THE TIME IN MINUTES BETWEEN THE INPUT (USUALLY
 ;        FROM INTERP) USED FOR GRID FDDA
 DIFTIM=720.,720.,0.,0.,0.,0.,0.,0.,0.,0.,      ; 3D ANALYSIS NUDGING
         180.,180.,0.,0.,0.,0.,0.,0.,0.,0.,      ; SFC ANALYSIS NUDGING
 ;
 ;        GRID NUDGE THE WIND FIELD? 0 = NO; 1 = YES
 IWIND=1,1,0,0,0,0,0,0,0,0,                 ; 3D ANALYSIS NUDGING
        1,1,0,0,0,0,0,0,0,0,                 ; SFC ANALYSIS NUDGING
 ;
 ;        NUDGING COEFFICIENT FOR WINDS ANALYSES
 GV=2.5E-4,1.0E-4,0.,0.,0.,0.,0.,0.,0.,0.,        ; 3D ANALYSIS NUDGING
    2.5e-4,1.0E-4,0.,0.,0.,0.,0.,0.,0.,0.,        ; SFC ANALYSIS NUDGING
 ;
 ;        GRID NUDGE THE TEMPERATURE FIELD? 0 = NO; 1 = YES
 ITEMP=1,1,0,0,0,0,0,0,0,0,                 ; 3D ANALYSIS NUDGING
        1,1,0,0,0,0,0,0,0,0,                 ; SFC ANALYSIS NUDGING
 ;
 ;        NUDGING COEFFICIENT FOR TEMPERATURE ANALYSES
 GT=2.5e-4,1.0E-4,0.,0.,0.,0.,0.,0.,0.,0.,        ; 3D ANALYSIS NUDGING
    2.5e-4,1.0E-4,0.,0.,0.,0.,0.,0.,0.,0.,        ; SFC ANALYSIS NUDGING
 ;
 IMOIS=1,1,0,0,0,0,0,0,0,0,                 ; 3D ANALYSIS NUDGING
       1,1,0,0,0,0,0,0,0,0,                  ; SFC ANALYSIS NUDGING
 ;
 ;        NUDGING COEFFICIENT FOR THE MIXING RATIO ANALYSES
 GQ=1.E-5,1.E-5,0.,0.,0.,0.,0.,0.,0.,0.,         ; 3D ANALYSIS NUDGING
    1.E-5,1.E-5,0.,0.,0.,0.,0.,0.,0.,0.,         ; SFC ANALYSIS NUDGING
 ;
 ;        GRID NUDGE THE ROTATIONAL WIND FIELD? 0 = NO; 1 = YES
 IROT=0,0,0,0,0,0,0,0,0,0,                  ; 3D ANALYSIS NUDGING
;
 ;        NUDGING COEFFICIENT FOR THE ROTATIONAL COMPONENT OF THE WINDS
 GR=5.E6,5.E6,0.,0.,0.,0.,0.,0.,0.,0.,           ; 3D ANALYSIS NUDGING
 ;
 ;         IF GRID NUDGING (I4D(1,1)=1) AND YOU WISH TO EXCLUDE THE
 ;         BOUNDARY LAYER FROM FDDA OF COARSE GRID THREE DIMENSIONAL
 ;         DATA (USUALLY FROM INTERP),
 ;              0 = NO,  INCLUDE BOUNDARY LAYER NUDGING
 ;              1 = YES, EXCLUDE BOUNDARY LAYER NUDGING
 INONBL =     0,0,0,0,0,0,0,0,0,0,          ; U WIND
              0,0,0,0,0,0,0,0,0,0,          ; V WIND
              0,0,0,0,0,0,0,0,0,0,          ; TEMPERATURE
              0,0,0,0,0,0,0,0,0,0,          ; MIXING RATIO
 ;
```

```
;         RADIUS OF INFLUENCE FOR SURFACE ANALYSIS (KM).
;         IF I4D(2,1)=1 OR I4D(2,2)=1, ETC, DEFINE RINBLW (KM) USED
;         IN SUBROUTINE BLW TO DETERMINE THE HORIZONTAL VARIABILITY
;         OF THE SURFACE-ANALYSIS NUDGING AS A FUNCTION OF SURFACE
;         DATA DENSITY.  OVER LAND, THE STRENGTH OF THE SURFACE-
;         ANALYSIS NUDGING IS LINEARLY DECREASED BY 80 PERCENT AT
;         THOSE GRID POINTS GREATER THAN RINBLW FROM AN OBSERVATION
;         TO ACCOUNT FOR DECREASED CONFIDENCE IN THE ANALYSIS
;         IN REGIONS NOT NEAR ANY OBSERVATIONS.
RINBLW=250.,
;
;         SET THE NUDGING PRINT FREQUENCY FOR SELECTED DIAGNOSTIC
;         PRINTS IN THE GRID (ANALYSIS) NUDGING CODE (IN CGM
;         TIMESTEPS)
NPFG=50,
;
;       **************** OBSERVATION NUDGING ***************
;
;
;      INDIVIDUAL OBSERVATION NUDGING.  VARIABLES THAT ARE ARRAYS
;      USE THE FIRST DIMENSION (COLUMN) AS THE DOMAIN IDENTIFIER:
;          COLUMN 1 = DOMAIN #1, COLUMN 2 = DOMAIN #2, ETC.
;
;        IS THIS INDIVIDUAL OBSERVATION NUDGING? 0 = NO; 1 = YES
I4DI   =0,0,0,0,0,0,0,0,0,0,
;
;        OBS NUDGE THE WIND FIELD FROM STATION DATA? 0 = NO; 1 = YES
ISWIND =1,0,0,0,0,0,0,0,0,0,
;
;        NUDGING COEFFICIENT FOR WINDS FROM STATION DATA
GIV  =4.E-4,4.E-4,0.,0.,0.,0.,0.,0.,0.,0.,
;
;        OBS NUDGE THE TEMPERATURE FIELD FROM STATION DATA? 0 = NO; 1 = YES
ISTEMP=1,0,0,0,0,0,0,0,0,0,
;
;        NUDGING COEFFICIENT FOR TEMPERATURES FROM STATION DATA
GIT  =4.E-4,4.E-4,0.,0.,0.,0.,0.,0.,0.,0.,
;
;        OBS NUDGE THE MIXING RATIO FIELD FROM STATION DATA? 0 = NO; 1 = YES
ISMOIS=1,0,0,0,0,0,0,0,0,0,
;
;        NUDGING COEFFICIENT FOR THE MIXING RATIO FROM STATION DATA
GIQ  =4.E-4,4.E-4,0.,0.,0.,0.,0.,0.,0.,0.,
;
;        THE OBS NUDGING RADIUS OF INFLUENCE IN THE
;        HORIZONTAL IN KM FOR CRESSMAN-TYPE DISTANCE-WEIGHTED
;        FUNCTIONS WHICH SPREAD THE OBS-NUDGING CORRECTION
;        IN THE HORIZONTAL.
RINXY=240.,
;
;        THE OBS NUDGING RADIUS OF INFLUENCE IN THE
;        VERTICAL IN SIGMA UNITS FOR CRESSMAN-TYPE DISTANCE-
;        WEIGHTED FUNCTIONS WHICH SPREAD THE OBS-NUDGING
;        CORRECTION IN THE VERTICAL.
RINSIG=0.001,
;
;        THE HALF-PERIOD OF THE TIME WINDOW, IN MINUTES, OVER
;        WHICH AN OBSERVATION WILL AFFECT THE FORECAST VIA OBS
;        NUDGING. THAT IS, THE OBS WILL INFLUENCE THE FORECAST
;        FROM TIMEOBS-TWINDO TO TIMEOBS+TWINDO.  THE TEMPORAL
;        WEIGHTING FUNCTION IS DEFINED SUCH THAT THE OBSERVATION
;        IS APPLIED WITH FULL STRENGTH WITHIN TWINDO/2. MINUTES
;        BEFORE OR AFTER THE OBSERVATION TIME, AND THEN LINEARLY
;        DECREASES TO ZERO TWINDO MINUTES BEFORE OR AFTER THE
;        OBSERVATION TIME.
TWINDO=40.0,
;
;        THE NUDGING PRINT FREQUENCY FOR SELECTED DIAGNOSTIC PRINT
;        IN THE OBS NUDGING CODE (IN CGM TIMESTEPS)
NPFI=20,
;
;        FREQUENCY (IN CGM TIMESTEPS) TO COMPUTE OBS NUDGING WEIGHTS
IONF=2,
IDYNIN=0,  ;for dynamic initialization using a ramp-down function to gradually
```

```
 ;          turn off the FDDA before the pure forecast, set idynin=1 [y=1, n=0]
 DTRAMP=60.,;the time period in minutes over which the
 ;          nudging (obs nudging and analysis nudging) is ramped down
 ;          from one to zero.  Set dtramp negative if FDDA is to be ramped
 ;          down BEFORE the end-of-data time (DATEND), and positive if the
 ;          FDDA ramp-down period extends beyond the end-of-data time.
 &END
EOF
#
########################################################################
####################    END USER MODIFICATION             ##################
####################                                      ##################
####################                                      ##################
########################################################################
#
#    This is where the utility programs/executable come from
#
set MesoUser = /fs/othrorgs/home0/mesouser/MM5V2/MM5
#
set LETTERS = (A B C D E F G H I J K L M N O P Q R S T U V W X Y Z)
#
if ( $compile == yes ) then
  make clean
  make
   set toast = $status
   if       ( $toast != 0 ) then
      echo "error in the compile, stopping"
      if ( -e mm5v2-debug.tar ) rm mm5v2-debug.tar
      tar -cf .mm5v2-debug.tar * ; mv .mm5v2-debug.tar mm5v2-debug.tar
      echo " mswrite -t $RetPd mm5v2-debug.tar $OutMM/mm5v2-debug.tar "
      mswrite -t $RetPd mm5v2-debug.tar $OutMM/mm5v2-debug.tar
      exit(1)
   else if ( $ENVIRONMENT != BATCH ) then
      echo -n "compile complete, continue?    (yes)    "
      set ans = "$<"
      if (( $ans == "n" ) || ( $ans == "no" )) then
         exit (1)
      endif
   endif
   if ( -e mm5exe.out ) rm mm5exe.out
   tar -cf .mm5exe.out * ; mv .mm5exe.out mm5exe.out
   mswrite mm5exe.out ${ExpName}/${CaseName}/mm5exe.tar
#
   if ( $execute == no ) exit (0)
else
#
#       Update information needed to create mmlif
#
   cc -o parseconfig Util/parseconfig.c
   ./parseconfig
endif
#
#  remove object files to avoid exceeding inode limit
#
make rm_obj
#
#  edit mmlif file to remove comments
#
make mmlif
cd Run
cp mmlif mmlif.comment
sed -f ../Util/no_comment.sed mmlif | grep "[A-Z, a-z]" > mmlif.tmp
mv mmlif.tmp mmlif
#
#       cd to Run, and run the model there
#
rm fparam lparam oparam pparam
#
#       get boundary conditions from MS
#
msread bdyout $InBdy
#
#       1. initial conditions
#
```

```
set NUMFIL = 1
while ( $NUMFIL <= ${#InMM} )
   msread fort.1$NUMFIL $InMM[${NUMFIL}]
   if (( $FDDAsw == Anly ) || ( $FDDAsw == Both )) then
      cp fort.1$NUMFIL fort.3$NUMFIL
   endif
   @ NUMFIL ++
end
#
#       2. input restart conditions
#
if ( $STARTsw == ReStart ) then
   set NUMFIL = 1
   while ( $NUMFIL <= ${#InRst} )
      msread fort.9$NUMFIL $InRst[${NUMFIL}]
      @ NUMFIL ++
   end
endif
#
#       3. get surface analyses files for nudging
#
if (( $FDDAsw == Anly ) || ( $FDDAsw == Both )) then
   foreach i ( $In4DSfc )
        set NUMFIL = `echo $i | sed 's/.*\(.\)/\1/'`
        echo "Current domain is $NUMFIL"
        msread fort.7$NUMFIL $i
   end
endif
#
#       4. observations if OBS nudging
#
if (( $FDDAsw == Obs ) || ( $FDDAsw == Both )) then
   foreach i ( $In4DObs )
        set NUMFIL = `echo $i | sed 's/.*\(.\)/\1/'`
        echo "Current domain is $NUMFIL"
        msread fort.6$NUMFIL $i
   end
endif
#
#       set up fortran input files for MM5
#
ln -s mmlif                          fort.10
ln -s ehtran                         fort.8
ln -s bdyout                         fort.9
ln -s landuse.tbl                        fort.19

#
ls -l
#
#       run MM5
#
date
size mm5.exe
./mm5.exe >&! mm5.print.out
cat mmlif.comment >> mm5.print.out
#
set toast = $status
if ( $toast != 0 ) then
   echo "error in the forecast, stopping"
   debug -s ./Run/mm5.exe
   if ( $ENVIRONMENT != BATCH ) exit(1)
endif
#
#       if interactive, probably do not want to dispose files
#
if ( $ENVIRONMENT != BATCH ) then
   echo -n "test mm5 run complete, continue?     (no)    "
   set ans = "$<"
   if (( $ans != "y" ) && ( $ans != "yes" )) then
      exit (0)
   endif
endif
#
#       print and save the print output
```

```
#
ja -s >! acct
cat acct >> mm5.print.out
if ( $ENVIRONMENT == BATCH ) cat mm5.print.out
#
#       history output   41-49
#       save file output 51-59
#       shutdown output  61-69
#
ls -ls
set Tens  = (   4     5     6   )
set Name  = ( MMOUT SAVE SHUTDO )
foreach OutType ( 1 2 3 )
   set OutFileType = $Name[$OutType]
   foreach Units ( 1 2 3 4 5 6 7 8 9 )
      if ( -e fort.$Tens[$OutType]${Units} ) then
         echo ls -l fort.$Tens[$OutType]${Units} >! hold
         set test = `source hold`
         if ( $test[5] < 1000000000 ) then
            mswrite -t $RetPd fort.$Tens[$OutType]$Units \
                    $OutMM/${OutFileType}_DOMAIN${Units}
         else
          if ($Tens[$OutType] == 4) then
            if ( ! -e split.csh ) then
            cp ${MesoUser}/split.csh .
            chmod +x split.csh
            endif
            mv fort.$Tens[$OutType]${Units} ${OutFileType}_DOMAIN${Units}
            split.csh ${OutFileType}_DOMAIN${Units}

            set Numb = 0
            foreach fil ( `ls mmtmp*` )
            @ Numb ++
            echo "mswrite $fil \
                  $OutMM/${OutFileType}_DOMAIN${Units}_$LETTERS[$Numb] "
            mswrite -t $RetPd $fil \
                  $OutMM/${OutFileType}_DOMAIN${Units}_$LETTERS[$Numb]
            end

            rm mmtmp*
          else

            bsplit -1000000000 fort.$Tens[$OutType]${Units} \
                   small.$Tens[$OutType]${Units}.
            set NUM = 1
            echo ls small.$Tens[$OutType]${Units}.* >! hold
            set test = `source hold`
            foreach split ( $test )
               mswrite -t $RetPd $split \
                  $OutMM/${OutFileType}_DOMAIN${Units}_BSPLIT_$LETTERS[$NUM]
               @ NUM ++
            end
          endif

         endif
      endif
   end
end
#
# tar the source code, objects, executable, and output together
# save the MM5 tar file on MSS
#
if ( $ENVIRONMENT == BATCH ) then
   rm mm5out.tar
   tar -cf .mm5out.tar mm5.print.out mmlif mm5.exe
   mv .mm5out.tar mm5out.tar
   echo " mswrite -t $RetPd mm5out.tar $OutMM/mm5out.tar "
   mswrite -t $RetPd mm5out.tar $OutMM/mm5out.tar
endif
ls -ls
```

D.11 graph.deck

```
# QSUB -r GRAPH                                # request name
# QSUB -q reg                                  # job queue class
# QSUB -eo                                     # stdout and stderr together
# QSUB -lM 10Mw                                # maximum memory
# QSUB -lT 2000                                # time limit
# QSUB                                         # no more qsub commands
#
ja
set echo
#
#
#
#
#             ************************************
#             ******    graph batch C shell    ******
#             ******         V2 System         ******
#             ******   for both cf77 and f90 ******
#             ************************************
#
#
#
if ( $?ENVIRONMENT ) then
   echo "environment variable defined as $ENVIRONMENT"
else
   setenv ENVIRONMENT INTERACTIVE
   echo "environment variable defined as $ENVIRONMENT"
endif
#
if ( $ENVIRONMENT == BATCH ) then
   cd $TMPDIR
endif
#
#       define MSS pathname
#
# this should be the user's case or experiment (for example TEST/EXP1/...).
# Note that there is no / in front of the pathname TEST.
#
    set ExpName = MM5V2/TEST/EXP1           # GRAPH input/output MSS pathname
    set RetPd   = 365                       # MSS file retention period in days
#
#       set MSS output path
#
    set OutGraph  =   $ExpName
#
#       locations for datasets
#
    set ID      =   1                       # input domain ID
    set InData  =   $ExpName/MMOUT_DOMAIN$ID
#
# if compile = yes, compile graph program
# Note : graph program dynamically allocates memory. Therefore,
#        if you don't have your own mods to the graph program, there
#        is no need to compile the graph program.
#        Unless your dataset dimension is greater than 100x100x50
#        In this case, edit the file data.incl and scratch.incl in the
#        src directory
#
    set compile = no
#   set compile = yes
#
#      Do you want to use your source code tar file?
#
if ( $compile == yes ) then
    set UseMySource = no
#   set UseMySource = yes
endif
#
#       define host name
# set up the remote host location where your source code may locate
#
    set Host    = username@host.domain:/usr/tmp/username
```

```
#
#    ------ GET GRAPH TAR FILE ---------------------------
#
if ( $compile == yes ) then
     if ( $UseMySource == yes ) then
         rcp $Host/graph.tar .
         tar -xf graph.tar
     else if ( ( $UseMySource == no ) && ( $ENVIRONMENT == BATCH ) ) then
         msread graph.tar.Z /MESOUSER/MM5V2/GRAPH/GRAPH.TAR.Z
#        cp /fs/othrorgs/home0/mesouser/MM5V2/Graph/graph.tar.Z .
         uncompress graph.tar
         tar -xf graph.tar
     endif
     rm graph.tar
endif
#
#
#    ------ GRAPH TABLES AND NAMELIST --------------------------
#
#    1. g_plots.tbl  -- need user editing
#
if ( -e g_plots.tbl ) rm g_plots.tbl
cat >! g_plots.tbl << "EOF"
TIME LEVELS (MDATE FORMAT YYMMDDHH): FROM 93031300 TO 93031400 BY 6
PRESSURE LEVEL MANDATORY: FROM 1000 TO PTOP
PRESSURE LEVEL NON STANDARD: FROM SFC TO PTOP by 3
SIGMA LEVEL: FROM 1 TO KMAX BY 1
THETA LEVEL: NONE
TITLE: STORM OF THE CENTURY - MM5V2 NH
-----------------------------------------------------------------------------
PLOT | FIELD  | UNITS  | CONTOUR  | SMOOTH || OVERLAY | UNITS  | CONTOUR  | SMOOTH
T/F  |        |        | INTERVAL | PASSES || FIELD   |        | INTERVAL | PASSES
-----|--------|--------|----------|--------||---------|--------|----------|------
 F   |  TER   |   m    |  100.    |   0    ||         |        |          |
 F   | LNDUS  |category|          |   0    ||         |        |          |
 T   |  PSLV  |   mb   |    2.    |   0    ||         |        |          |
 F   | RAINN  |   cm   |    2.    |   0    ||         |        |          |
 F   | RAINC  |   cm   |    1.    |   0    ||         |        |          |
 F   | RAINT  |   cm   |    1.    |   0    ||         |        |          |
 F   | RTENDN |   mm   |    5.    |   0    ||         |        |          |
 F   | RTENDC |   mm   |    5.    |   0    ||         |        |          |
 F   | RTENDT |   mm   |    5.    |   0    ||         |        |          |
 F   | PRH2O  |   mm   |    1.    |   0    ||         |        |          |
 F   |  PSFC  |   mb   |   12.    |   0    ||         |        |          |
 F   | PTEND  |   mb   |    1.    |   0    ||         |        |          |
 F   | TGD    |   C    |    4.    |   0    ||         |        |          |
 F   | OBS/BOX|stations|    2.    |   0    ||         |        |          |
 F   | WIND   |  m/s   |    5     |   0    ||  VECT   |  m/s   |    1.    |    0
 T   | WIND   |  m/s   |    5     |   0    ||  BARB   |  m/s   |    1.    |    0
 F   | BARBG  |  m/s   |    4.    |   0    ||  WINDG  |  m/s   |    1.    |    0
 T   | T      |   C    |    2.    |   0    ||         |        |          |
 F   |  QV    |  g/kg  |    2.    |   0    ||         |        |          |
 T   |  RH    |   %    |   10.    |   0    ||         |        |          |
 F   | CLW    |  g/kg  |    1.    |   0    ||         |        |          |
 F   | RNW    |  g/kg  |    1.    |   0    ||         |        |          |
 F   | ICLW   |   cm   |    1.    |   0    ||         |        |          |
 F   | IRNW   |   cm   |    1.    |   0    ||         |        |          |
 F   | PRH2O  |   cm   |    1.    |   0    ||         |        |          |
 F   | SNOW   |  g/kg  |    1.    |   0    ||         |        |          |
 F   | ICE    |  g/kg  |    1.    |   0    ||         |        |          |
 F   | NICE   |  g/kg  |    1.    |   0    ||         |        |          |
 F   | HEIGHT |   m    |   40.    |   0    ||         |        |          |
 F   | HEIGHT |   m    |   40.    |   0    ||  T      |   C    |    3.    |    0
 F   | VOR    |10**5/s |   10.    |   0    ||         |        |          |
 F   | OMG    |  ub/s  |    2.    |   0    ||         |        |          |
 F   | H      |  m     |   40.    |   0    ||         |        |          |
 F   | U      |   m/s  |   10.    |   0    ||         |        |          |
 F   | V      |   m/s  |   10.    |   0    ||         |        |          |
 F   | W      |  cm/s  |   10.    |   0    ||         |        |          |
 F   | VAB    |10**5/s |   10.    |   0    ||         |        |          |
 F   | DIV    |  1/s   |    0.    |   0    ||         |        |          |
 F   | MDIV   |  1/s   |    0.    |   0    ||         |        |          |
 F   | QVEC   |  k/ms  |    0.    |   0    ||         |        |          |
 F   | QDIV   | k/m^2s |    0.    |   0    ||         |        |          |
```

```
 F   | P      | mb      |    4.    |    0    ||         |       |          |
 F   | PP     | mb      |    1.    |    0    ||         |       |          |
 F   | TD     | C       |    2.    |    0    ||         |       |          |
 F   | TDD    | C       |    2.    |    0    ||         |       |          |
 F   | THETA  | K       |    4.    |    0    ||         |       |          |
 F   | THETAE | K       |    4.    |    0    ||         |       |          |
 F   | LI     | K       |    2.    |    0    ||         |       |          |
 F   | MSE    | j/kg    |    1.    |    0    ||         |       |          |
 F   | MSS    | j/kg    |    1.    |    0    ||         |       |          |
 F   | PV     |Km/s^3Pa|    1.    |    0    ||         |       |          |
 F   | BARB   | m/s     |    0.    |    0    ||         |       |          |
 F   | VECT   | m/s     |    0.    |    0    ||         |       |          |
 F   | VESL   | m/s     |   10.    |    0    ||         |       |          |
 F   | BARBG  | m/s     |    0.    |    0    ||         |       |          |
 F   |   8    |  6      |    20    |   20    ||         |       |          |
 F   | THETA  |  K      |    4.    |    0    ||   CXW   | cm/s  |    6     |  0
 F   |   8    |  6      |    20    |   20    ||         |       |          |
 F   | THETA  |  K      |    4.    |    0    ||    W    | cm/s  |    1.    |  0
 F   |   8    |  6      |    20    |   20    ||         |       |          |
 F   | THETAE |  K      |    4.    |    0    ||   AM    | m/s   |    5.    |  0
 FI304 | P      | mb      |     4.   |    0    ||         |       |          |
 FP500 | THETA  | K       |     4.   |    0    ||         |       |          |
 FI300 | PV     |Km/s^3Pa |     1.   |    0    ||         |       |          |
 FP700 | H      | m       |    40.   |    0    ||         |       |          |
 FP700 | T      |  C      |     2.   |    0    ||         |       |          |
 FI304 | WIND   |  m/s    |     5    |    0    ||  BARB   |  m/s  |    4.    |  0
 T    |SKEWTXY| A                   |   15    |   20    ||    |   | |
--------------------------------------------------------------------------
"EOF"
#
#    2. g_defaults.nml  -- namelist for tailoring plots
#       include the following for plotting subdomains
#      ISTART=1,       ; beginning I of a sub-domain
#      IEND=35,        ; ending I of a sub-domain
#      JSTART=1,       ; beginning J of a sub-domain
#      JEND=41,        ; ending J of a sub-domain
#
if ( -e g_defaults.nml ) rm g_defaults.nml
cat >! g_defaults.nml << "EOF"
 &JOEDEF
      LW1=2000,        ; line width in increment of 1000
      LW2=2000,
      DASH1=-682,      ; dash pattern, standard NCAR Graphics
      DASH2=-682,      ; 4092, 3640, 2730, -682
      COLOR1=12,       ; color code for first plot
      COLOR2=9,
      COLOR3=8,
      COLOR4=12,       ; color code for overlay plot
      COLOR5=9,
      COLOR6=8,
      HDRINFO=F,       ; true=print header and stop
      LOGP=0,          ; 0=linear in p, 1=linear in ln p
      XPTOP=100.,      ; top of cross section plots (mb)
      ICONDRV=F,F,     ; use CONDRV or CONREC
      LMETH=-2,        ; if condrv, which contouring method
      LEVELS=0.4,
      LABLINE=1,       ; 0: no contour line labels
      LABMESG=0,       ; 1: no message below conrec plot
      NOZERO=0,        ; 0: allow zero line; 1: no min/max zero line; 2: no zero line
      IHIRES=0,        ; 1: use high resolution US county line/China coastline
 &END ;
"EOF"
#
#    3. g_map.tbl  -- change only if needed
#
if ( -e g_map.tbl ) rm g_map.tbl
cat > g_map.tbl << "EOF"
-----------------------------------------------------------------------------
                                 MAP DETAILS
LL | DASH | INT | LB | LSZ | LQL | P | TTL | TSZ | TQL | OUT | DOT | LW   | SP
-----------------------------------------------------------------------------
A  | TI   | D   | M  | 14  | 00  | Y | Y   | 12  |  0  | PS  | N   | 1    |
-----------------------------------------------------------------------------
                                  MAP COLORS
```

```
LL LINES | LABELS | TITLE  | STATES     | COUNTRIES  | CONTINENTS | PERIMETER
------------------------------------------------------------------------------
1        | 1      |        | 1          | 1          | 1          | 1
------------------------------------------------------------------------------
"EOF"
#
#    4. g_map.tbl  -- change only if needed
#
if ( -e g_color.tbl ) rm g_color.tbl
cat > g_color.tbl << "EOF"
------------------------------------------------------------------------------
                                 COLOR TABLE
COLOR                  | RED  | GREEN | BLUE | NUMBER
------------------------------------------------------------------------------
WHITE                  | 1.00 | 1.00  | 1.00 | 1
LIGHT GRAY             | 0.66 | 0.66  | 0.66 | 2
DARK GRAY              | 0.40 | 0.40  | 0.40 | 3
BLACK                  | 0.00 | 0.00  | 0.00 | 4
SKY BLUE               | 0.20 | 0.56  | 0.80 | 5
BLUE                   | 0.00 | 0.00  | 1.00 | 6
LIGHT YELLOW           | 0.80 | 0.80  | 0.00 | 7
MAGENTA                | 1.00 | 0.00  | 1.00 | 8
YELLOW                 | 1.00 | 1.00  | 0.00 | 9
GREEN                  | 0.00 | 1.00  | 0.00 | 10
FOREST GREEN           | 0.14 | 0.25  | 0.14 | 11
CYAN                   | 0.00 | 1.00  | 1.00 | 12
TAN                    | 0.40 | 0.30  | 0.20 | 13
BROWN                  | 0.25 | 0.20  | 0.15 | 14
ORANGE                 | 1.00 | 0.50  | 0.00 | 15
RED                    | 1.00 | 0.00  | 0.00 | 16
MID-BLUE               | 0.00 | 0.50  | 1.00 | 17
DULL MID-BLUE          | 0.00 | 0.15  | 0.30 | 18
BRIGHT FOREST GREEN    | 0.20 | 0.40  | 0.20 | 19
DULL ORANGE            | 0.60 | 0.30  | 0.00 | 20
------------------------------------------------------------------------------
"EOF"
########################################################################
####################                                  ##################
####################    END USER MODIFICATION         ##################
####################                                  ##################
########################################################################
#
#
#    This is where the standard source code/executable come from
#
set MesoUser = /fs/othrorgs/home0/mesouser/MM5V2/Graph
#
#    The make rules are contained in the file configure.make
#
# --------- create graph executable -------------------
#
set LETTERS = (A B C D E F G H I J K L M N O P Q R S T U V W X Y Z)
set DomId = (1 2 3 4 5 6 7 8 9 10)
#
#set MACH = `uname -a | awk '{ print $2 }'`
if ( $compile == yes ) then
      echo "using configure.make file with f90 option"
      cp Templates/configure.make.f90 configure.make
      msread src/plots.o /MESOUSER/plots/object/f90/plots.o
make
   set toast = $status
   if ( $ENVIRONMENT != BATCH ) then
      echo -n "compile complete, continue?    (yes)   "
      set ans = "$<"
      if (( $ans == "n" ) || ( $ans == "no" )) then
         exit (1)
      endif
   else if (( $ENVIRONMENT == BATCH ) && ( $toast != 0 )) then
      echo "error in the compile, stopping"
      exit(1)
   endif
else
     echo "using configure.make file with f90 option"
      msread graph.exe /MESOUSER/MM5V2/GRAPH/GRAPH.EXE
```

```
#     cp /fs/othrorgs/home0/mesouser/MM5V2/Graph/graph.exe.Z src/graph.exe.Z
#     uncompress src/graph.exe
endif
#
if ( ! -e graph.exe ) ln -s src/graph.exe graph.exe
#
#  edit namelist to remove comments
#
sed -f ./Templates/no_comment.sed g_defaults.nml > mif.tmp
mv mif.tmp g_defaults.nml
#
# --------- get data sets from MS
#
set NUMFIL = 1
while ( $NUMFIL <= ${#InData} )
   set Local  = graph$LETTERS[${NUMFIL}]
   set Remote = $InData[${NUMFIL}]
   msread $Local $Remote
   @ NUMFIL ++
end
#
# --------- run GRAPH
#    graph.csh can also be obtained from ~mesouser/MM5V2/Graph
#       or graph.tar.Z file
#
cp $MesoUser/graph.csh .
chmod +x graph.csh
#
date
if      ( ${#InData} == 1 ) then
   graph.csh 1 1 graphA >& graph.print.out
else if ( ${#InData}  > 1 ) then
   graph.csh 1 ${#InData} graph >& graph.print.out
endif
#
set toast = $status
if (( $ENVIRONMENT == BATCH ) && ( $toast != 0 )) then
   cat graph.print.out
   echo "error in graph, stopping"
   rm core
   exit(1)
endif
#
if ( -e acct ) rm acct
ja -s >! acct
cat acct >> graph.print.out
if ( $ENVIRONMENT == BATCH ) cat graph.print.out
#
#                graph output - meta code file
#
echo "mswrite -t $RetPd gmeta $OutGraph/PLOT_DOMAIN$ID"
mswrite -t $RetPd gmeta $OutGraph/PLOT_DOMAIN$ID
#
ls -ls
ja -st
```