I.2-UNIX-DHM PROGRAM EXECUTION INFORMATION FOR DISTRIBUTED HYDROLOGIC MODELING (DHM) ON UNIX SYSTEMS

Contents

Background

DHM is the set of features integrated into NWSRFS for distributed hydrologic modeling. The primary component is DHM-OP, an NWSRFS operation (#64), which executes in batch mode through OFS-FCST and in GUI mode through IFP. The existing dependencies for executing operations in OFS-FCST and IFP (e.g. tokens, scripts and executables) still apply. New dependencies for the DHM-OP include Java archives and shared libraries described in more detail below.

DHM-OP uses the FS5files database to read and write timeseries data and XMRG files for gridded input and output. OFS-FCST can also write netCdf grids for display in D2D

Apps Defaults Tokens

dhm_data_dir: $(ofs_files)/$(ofs_level)/dhmdata
root directory location for DHM input/output xmrg files

Before running DHM for the first time, this directory and its subdirectories need to be created using the script create_files_group.

create_files_group **group_name**
(group name should be equal to $ofs_level)

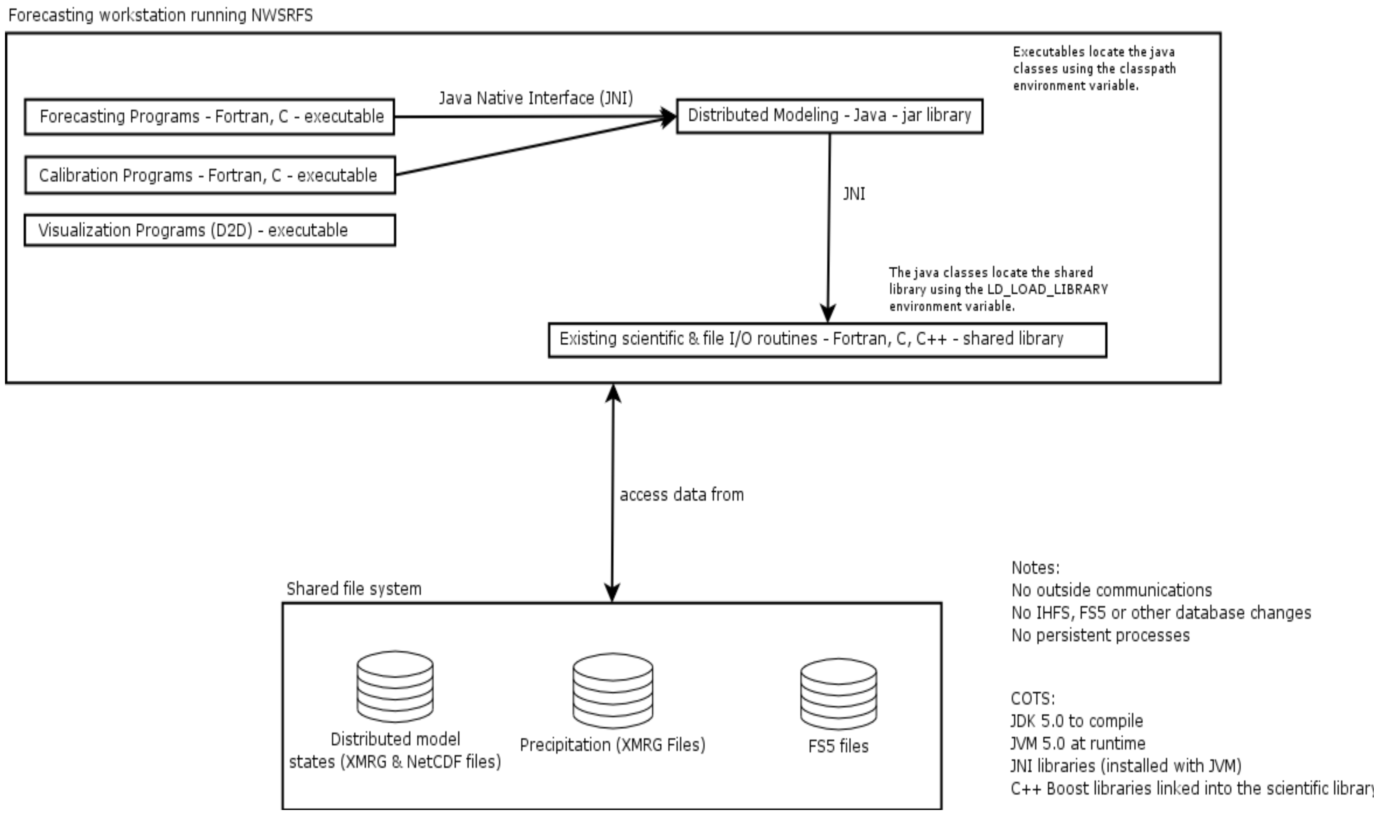
ifp_dhm_data_dir: /data/dhm/$(LOGNAME)
root directory for location for DHM input/output xmrg files when running DHM through IFP.
ifp_griddb_dir: $(ifp_dhm_data_dir)/precip
directory with precip xmrgs for running DHM through IFP.

** Performance tests have shown this should be a directory local to the workstation running IFP

dhm_d2d_notify_bin_dir: /awips/fxa/bin
directory location for D2D executable
dhm_d2d_data_dir: /data/fxa/Grid/LOCAL/netCDF/DHM

directory location for D2D template and for DHM to write grid output file that will be displayed by D2D program.

rdhm_input_dir: $(geo_data)
directory location for grids and connectivity file used by RDHM.
dhm_rain_plus_melt_data_dir: $(geo_data)
directory location for rain plus melt grids gener.
turn_off_jvm: 1



trouble-shoot purpose; set to 1 to disable JVM – can't define DHM operation in segment definition if set this token to 1.

DHM Architecture

DHM-OP uses FORTRAN, C, C++ and Java. Unlike traditional NWSRFS operations, the modeling logic is not built into OFS-FCST. The NWSRFS code specific to DHM-OP (operation #64) is used as an adapter to various libraries with the modeling logic. The following schematic is an overview of the architecture.

DHM Libraries:

DHM shared (*.so) and java archive libraries (*.jar) are stored in the directory pointed to by the Apps_defaults token **util_rls**. DHM also uses JAVA system libraries in subdirectories to the token pointed to by the Apps_defaults token **sys_java_dir**. The shared libraries are accessed at run-time by adding these directories to the LD_LIBRARY_PATH environment variable. The JAVA archives (jars) are accessed by adding each jar file to the CLASSPATH environment variable. The current list of shared and jar libraries are:

Table 1: DHM Jar Files

Jar Name	Description
Open Source	
commons-io1.3.1.jar	http://jakarta.apache.org/commons/
toolsUI-2.2.12.jar	http://www.unidata.ucar.edu/software/netcdf-java/
jgrapht-0.6.0.jar	http://jgrapht.sourceforge.net/
commons-collections-3.1.jar	http://jakarta.apache.org/commons/collections/
OHD Developed	
dhm.jar	Store, load, and model DHM grid data
dhm-tests.jar	Junit and fit tests for dhm.jar
ofs.jar	Interface between NWSRSF and dhm
dhmguis.jar	Java guis for defining mods, editing grids, and choosing grid output
rdhmutilites.jar	Java wrappers for RDHM C++ code used to traverse connectivity file and read/write xmrg-like grids

Table 2: DHM Shared Libraries

Shared Library Name	Description
Java System libraries	
libjava.so	Java JVM library
libjvm.so	Java JVM library
libverify.so	Java JVM library
libhpi.so	Java JNI library
OHD Developed	
librdhmutilities.so	C++ library with utilities to traverse connectivity file and read/write xmrg-like grids
libdistrouting.so	Fortran and C code with Kinematic Routing algorithm

DHM Scripts:
The ofs script was updated to add the shared libraries to the LD_LIBRARY_PATH and java archive libraries to the CLASSPATH for running DHM through OFS-FCST (batch mode). A new ifp script, with the same purpose, was added for running DHM through IFP (GUI mode). The complete list of NWSRFS scripts used by DHM is:

Script Name	Description
ofs	Add dhm libraries to CLASSPATH and LD_LIBRARY_PATH for batch mode runs through FCST
start_ifp_nwsrfs	Add dhm libraries to CLASSPATH and LD_LIBRARY_PATH for gui mode runs through IFP
abort_nwsrfs	Cleans up log files generated by Java Virtual machine when FCST or IFP abnormally terminates
ifp_cleanup	Removes the temporary copy of dhm grids under /data/dhm/$LOGNAME created when running DHM through IFP
run_gridoutput_configuration	Starts a Java dialog box for choosing which grids to create when running DHM through IFP
run_dhm_grid_editor	Starts a utility used to convert grid data from RDHM format to DHM Format see http://www.nws.noaa.gov/oh/hrl/general/indexdoc.htm (link to gridEditor user's manual)

DHM Data

XMRG Input Grids:

The coupled Sacramento SMA and Kinematic Routing models use the grids in the following tables. The precipitation grids are standard XMRG files, located in the directory referenced by the **ofs_griddb_dir** (for running DHM from FCST) and **ifp_griddb_dir** (for running DHM from IFP) Apps_defaults tokens. The potential evapotranspiration, model parameter, and model state grids are gnu zipped files in an XMRG-like format. These files are located in subdirectories (pet, parameters, and states respectively) of the directory referenced by the **dhm_data_dir** Apps_defaults token. Each user ($ofs_level) has their own set of files. The parent DHM directory and its subdirectories are automatically created by running the create_files_group script.

The first record of the XMRG-like files is the same as standard XMRGs and consists of four integers:

- X- and Y-origins in HRAP coordinates, and
- numbers of columns and rows in the file.

The second record differs from standard XMRG files. It consists of 4 variables:

- an integer scale factor to convert integer*2 values into real values (to get the true values, divide the value stored in the file by this factor),
- and number of bytes per value (2 for integer*2 format and 4 for real*4 format).
- cell size in HRAP coordinates (1)
- the grids NODATA value (-999.)

As with standard XMRGs, data values are written with the southernmost row written to the file first and the northernmost row written last.

I. Sacramento SMA Parameter Grids

sac_LZFPM.gz [mm]	sac_UZK.gz [per day]
sac_LZFSM.gz [mm]	sac_UZTWM.gz [mm]
sac_LZPK.gz [per day]	sac_ZPERC.gz
sac_LZSK.gz [per day]	sac_PCTIM.gz [%]
sac_LZTWM.gz [mm]	sac_ADIMP.gz [%]
sac_PFREE.gz	sac_RIVA.gz
sac_REXP.gz	sac_SIDE.gz
sac_UZFWM.gz [mm]	sac_RSERV.gz

*Sacramento SMA parameter grids must exist in **$dhm_data_dir/parameters**

II. Sacramento SMA State Grids [a]

sac_uztwc_initial.mmddyyyyhhz.gz [mm]
sac_uzfwc_initial.mmddyyyhhz.gz [mm]
sac_lztwc_initial.mmddyyyyhhz.gz [mm]
sac_lzfpc_initial.mmddyyyyhhz.gz [mm]
sac_lzfsc_initial.mmddyyyyhhz.gz [mm]
sac_adimpc_initial.mmddyyyyhhz.gz [mm]

*Sacramento SMA state grids must exist in **$dhm_data_dir/states**

III. Kinematic Routing Parameter Grids

rutpix_DS.gz	Drainage Density
rutpix_ROUGH.gz	Hillslope Roughness
rutpix_SLOPH.gz	Hillslope Slope
rutpix_Q0CHN.gz	Coefficient in channel’s Kinematic Routing Equation
rutpix_QMCHN.gz	Exponent in channel’s Kinematic Routing Equation

Kinematic Routing parameter grids must exist in **$dhm_data_dir/parameters**

Kinematic Routing state grids must exist in **$dhm_data_dir/states**

IV. Kinematic Routing Initial State Grids [a]

rutpix_depth_initial.mmddyyyyhhz.gz	Depth of water in hillslope [mm]
rutpix_areacN_initial.mmddyyyyhhz.gz[b]	Cross-sectional area of water in channel

a = mmddyyyyhh represents the date and time for the saved state
b = N represents the channel reach number, there are usually 3 per 4km HRAP grid

Potential Evapotranspiration grids must exist in **$dhm_data_dir/pet**

V. Potential Evapotranspiration Rate on 16th day of month[c] [mm/day]

pe_JAN.gz	pe_JUL.gz
pe_FEB.gz	pe_AUG.gz
pe_MAR.gz	pe_SEP.gz
pe_APR.gz	pe_OCT.gz
pe_MAY.gz	pe_NOV.gz
pe_JUN.gz	pe_DEC.gz

[c] The DHM monthly pe grids are include the effects of the pe_adjustemnt grids

XMRG Output Grids:

OFS-FCST

When running DHM through OFS/FCST the results of the SAC-SMA and Kinematic Routing calculations (channel flow at each pixel in the modeled basin) is stored in a gnu-zipped XMRG-like file in a directory named channelflow (a subdirectory to the directory referenced by the **dhm_data_dir** Apps_defaults token).

VI. Channel Flow Grids [c]

channelflow.xmrg.mmddyyyyhh.gz [m^3/s]

c = mmddyyyyhh represent the date the saved state date and time

Also, during carryover save runs (OFS-FCST) the state grids mentioned in the above tables are created.

netCDF Grids:

netCDF grids are created during batch carryover save runs (OFS-FCST). The grids will contain the results for all basins modeled with a DHM-OP in the current carryover group and any previously modeled carryover groups.

The netCDF grid is stored as a time-stamped file (yyyymmdd_hhmm), where the date and time represent the model run-time (e.g. TODAY at 12Z). The file can have data for as far back as 5 days ago and 5 days forward. The amount of data is controlled using the carryover save dates.

For example if the following carryover save FCST deck is run on 1/6/2006. DHM will create a file named 20060101_1200 for all grid variables. Each file will contain data for days 0102 12Z to 0106 12Z.

```
@SETOPT
STARTRUN *-05
ENDRUN *
NUMCOSAV *-04 *-03 *-02 *-01 *
FUTPRECP(0)
@COMP FCEXEC
```

DHM netCDF Grid File *
SAC SMA Additional Impervious Area Contents [ADIMC, % full]
Lower Zone Free Secondary Contents [LZFPC, % full]
Lower Zone Tension Water Contents [LZFSC, % full]
Lower Zone Tension Water Contents [LZTWC, % full]
Upper Zone Free Water Contents [UZFWC, % full]
Upper Zone Tension Water Contents [UZTWC, % full]
Hillslope Water Depth [WATER_DEPTH, inches]
Channel flow (LOW Range) [ROUTED_FLOW, cfs]
Channel flow (MID Range) [ROUTED_FLOW_M, cfs]
Channel flow (HIGH Range) [ROUTED_FLOW_H, cfs]
Channel flow (ALL Ranges) [ROUTED_FLOW_C, cfs]

* one file containing all variables and times

IFP

When running DHM through IFP, the root directory for output grids is defined by the Apps_defaults token **ifp_dhm_data_dir.** There can be up to three subdirectories (channelflow, states, runoff) with the following data

VII. Channel Flow Grids *

channelflow.xmrg_mmddyyyhhz.mmddyyyyhh.gz [m^3/s]

* = the first mmddyyyyhh string represents the date for the start of the run; the second mmddyyyyhhz string represents the date for the data values

VIII. State Grids *

sac_VARIABLE_mmddyyyyhhz.mmddyyyyhhz.gz [mm]
rutpix_depth_mmddyyyyhhz.mmddyyyyhhz.gz [mm]
rutpix_areacN_mmddyyyyhhz.mmddyyyyhhz.gz [m^2]

* the first mmddyyyyhh string represents the date for the start of the run;
  the second mmddyyyyhhz string represents the date for the data values;
  VARIABLE represents the sac state (uztwc, uzfwc, lztwc, lzfsc, lzfpc, and adimpc)
  N represents the channel reach number, there are usually 3 per 4km HRAP grid

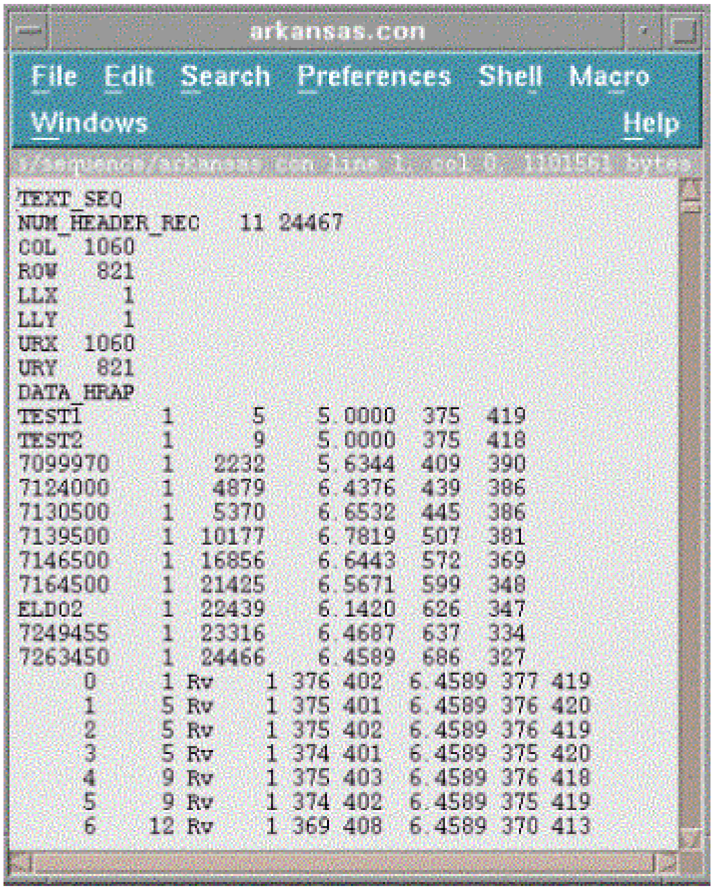
IX. Runoff Grids *

total_runoff_mmddyyyhhz.mmddyyyyhh.gz [mm]

* = the first mmddyyyyhh string represents the date for the start of the run; the second mmddyyyyhhz string represents the date for the data values

Connectivity file:

The connectivity file, with filename Connectivity, is a text file located in the directory referenced by **dhm_data_dir** Apps_defaults token. It stores an ordered list of all the cells in the model domain. The cells are ordered so that each time the next cell in the list is reached, computations for all upstream cells have already been completed. In the header of the connectivity file, the user specifies basin outlet cells and IDs of interest for a region. The same outlet IDs used in the connectivity file are also used in the fcinit input deck to choose where hydrographs will be generated. Figure 2 shows a screenshot of the top of a connectivity file.




```
TEXT_SEQ
NUM_HEADER_REC   11 24467
COL   1060
ROW    821
LLX      1
LLY      1
URX   1060
URY    821
DATA_HRAP
TEST1       1       5     5.0000   375   419
TEST2       1       9     5.0000   375   418
7099970     1    2232     5.6344   409   390
7124000     1    4879     6.4376   439   386
7130500     1    5370     6.6532   445   386
7139500     1   10177     6.7819   507   381
7146500     1   16856     6.6443   572   369
7164500     1   21425     6.5671   599   348
ELDO2       1   22439     6.1420   626   347
7249455     1   23316     6.4687   637   334
7263450     1   24466     6.4589   686   327
      0      1 Rv     1 376 402   6.4589 377 419
      1      5 Rv     1 375 401   6.4589 376 420
      2      5 Rv     1 375 402   6.4589 376 419
      3      5 Rv     1 374 401   6.4589 375 420
      4      9 Rv     1 375 403   6.4589 376 418
      5      9 Rv     1 374 402   6.4589 375 419
      6     12 Rv     1 369 408   6.4589 370 413
```

Figure 2 Example Connectivity File

Adding outlets to the connectivity file:

The connectivity file contains 9 lines of general header information, followed by a list of basin outlets defined for the model, followed by an ordered list of pixels covering the entire modeling domain. The top portion of a connectivity file is shown here.

```
TEXT_SEQ
NUM_HEADER_REC   2 24467
COL   1060
ROW    821
LLX      1
LLY      1
URX   1060
URY    821
DATA_HRAP
TEST1       1       5     5.0000   375   419
TEST2       1       9     5.0000   375   418
      0      1 Rv     1 376 402   6.4589 377 419
      1      5 Rv     1 375 401   6.4589 376 420
      2      5 Rv     1 375 402   6.4589 376 419
      3      5 Rv     1 374 401   6.4589 375 420
```

```
        4     9 Rv    1 375 403  6.4589 376 418
        5     9 Rv    1 374 402  6.4589 375 419
        6    12 Rv    1 369 408  6.4589 370 413
        7    12 Rv    1 369 406  6.4589 370 415
        8    13 Rv    1 375 404  6.4589 376 417
   .          .          .
   .          .          .
   .          .          .
```

Line 2: Lists the number of outlets identified in this file (2) and the total number of cells listed in the file (24467)
Lines 3, 4: Number of columns and rows. **
Lines 5, 6: Lower-left X and Y coordinates. **
Lines 7, 8: Upper-right X and Y coordinates. **
Lines 10-11: Basin outlet specifications. Information included in each line:

Col. 1: ASCII identifier. Must be the same as identifier used in FCINIT Input Deck.
Col. 2: ignore
Col. 3: Cell line number of this outlet
Col. 4: Representative area (mi^2) of cells in this basin. Computed as the USGS area divided by the number of HRAP cells in the basin. (program does not currently use this value)
Col. 5: hrapx coordinate of outlet cell (program does not currently use this value)
Col. 6: hrapy coordinate of outlet cell (program does not currently use this value)

Lines 12-24479: Cell List

Col. 1: Cell line number
Col. 2: Cell line number of the next downstream cell
Col. 3: ignore
Col. 4: ignore
Cols. 5-6: Local reference coordinates
Col. 7: Area represented by the cell in mi^2

Cols. 8-9: Globally referenced Hrapx and Hrapy (the true hrapx, hrapy coordinates)

To add a new outlet point, a new basin outlet specification line must be added (e.g. lines 10 – 11 in the example). As defined above, the information required for the added line includes 6 values: a basin identifier of eight characters or less, a dummy value of 1, the cell line number in the connectivity file corresponding to the basin outlet, a representative cell area, and the HRAP x and y coordinates of the outlet cell for this point. These values should all be entered on one line in free format, separated by at least one space.

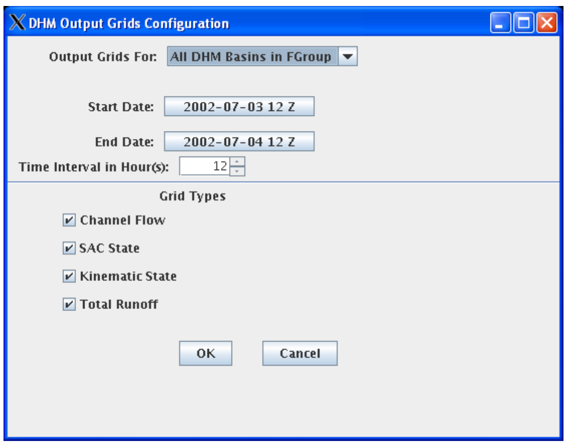
To determine the cell line number corresponding to a new outlet point, do a text string search on the connectivity file for the true hrapx and hrapy coordinates of the point you are adding (separated by a single space). For example, search for “375 419” to identify the outlet line number for TEST1 in the above example. The search should find “375 419” in Columns 8-9 somewhere in the Cell List (lines 23-24489). The value in Column 1 of the cell line number is the third number in your basin specification line (e.g. 5 for TEST1 in the example). The fourth number, a representative cell area, can be taken from column 7 of the cell line number found in your search. _bottomMiscellaneous ~back from Tables_top

** - these values from the connectivity file are not used to determine the RFC's boundary. Instead the boundary is defined by the coordinates in the file located using the Appsdefaults tokens geo_data_dir and ifp_rfc.

 $geo_data_dir/$ifp_rfc/ascii/coord_$ifp_rfc.dat

Requesting Grid Output From IFP

When running DHM through IFP, grid output is requested using the following dialog (accessed from the Options menu)



From this dialog, the user enters three types of data:

1. Persistence for grid output
2. Timing Information
3. Grid output types

Persistence for Grid Output

Grid output persistence is defined from the **Output Grids For** drop-down box. The options are (All DHM Basins in FGROUP, Next DHM Basin in FGROUP, NONE). If All… is chosen DHM grids will be created when processing any segment in the FGROUP with a DHM-OP. When Next…. is chosen the next DHM-OP processed during the current FGROUP run will have grid output, and NONE turns off grid output

Timing Information

Which dates to write grids for is controlled with the **Start Date**, **End Date**, and **Time Interval in Hour(s)** options. Grid output will be created from Start Date to End Date at the specified Time Interval. For example if the Start Date is 07/01/2002 12z and the End Date is 07/02/2002 12z and the Time Interval in Hour(s) is 12. Two grids are created, one at 07/02/2002 00z and 07/02/2002 12z.

Grid output types

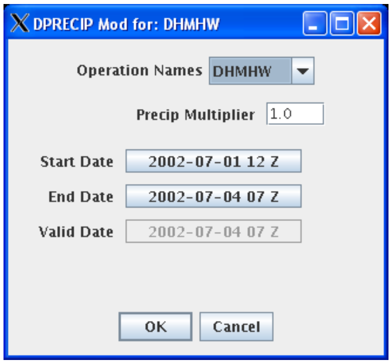
One or more types of grids to output are defined by checking the Grid Type box.

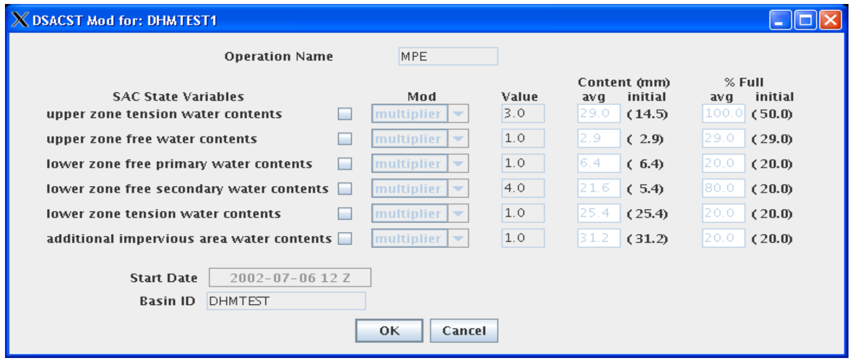
DHM Mods

Run-time modifications are supported by DHM-OP in batch (OFS-FCST) and GUI (IFP) mode. The supported mods are:

Mod Name	**Needed Inputs**	**Description**
DSACST	Valid Date – date the SAC carryover mod is applied [defaults to Start of the Run] KeyWord/Value pairs – 1 to 6 pairs of Keyword and value. Valid keywords are: • UZTWCM or UZTWCP • UZFWCM or UZFWCP • LZTWCM or LZTWCP • LZFSCM or LZFSCP • LZFPCM or LZFPCP • ADIMCM or ADIMCP	In the case of SAC State multiplier mods, the sac state in each grid cell in the basin (on the date given) is multiplied by the mod value In the case of SAC State percent full mods, the sac state in each grid cell is set to a percentage of the corresponding maximum state parameter. • UZTWCP uses UZTWM • UZFWCP uses UZFWM • LZTWCP uses LZTWM • LZFSCP uses LZFSM • LZFPCP uses LZFPM • ADIMCP uses UZTWM + LZTWM
DPRECIP	Start Date – Date the mod starts End Date – Date the mod ends Valid Date – Date after which the mod expires Multiplier – value used to lower/increases precipitation in each grid cell	The recorded XMRG in each grid cell for a given basin is lowered/increased by the multiplier value. The modifications are applied for every hourly grid within the Start Date to End Date window

Mods are entered in IFP using the following GUIs.





DHM Precip Mod GUI **DHM SAC state Mod GUI**

XXXXX NEED TO DESCRIBE WHAT COLORS MEAN IN SAC-STATE GUI

DHM Use of QPF for OB 8.3

DHM uses 6 hour QPF accumulation grids in XMRG format. The naming convention used is as follows:

xmrgPP_YYYYMMDDHHfhhh

PP is the increment -- 6

YYYYMMDDHH is the run date. For example, 2002041512 is the 12z QPF from April 15, 2002.

- **hhh** is the ending period. DHM uses 6-hour QPF, hhh can be 006 (period 1), 012 (period 2), 018 (period 3) or 024 (period 4), etc.

For example the grid xmrg6_2002041512f006 is the 6 hour QPF issued on April 15, 2002 12z for the period April 15, 2002 12z to April 15, 2002 18z

The steps for calculating QPF are:

1. Read the most recent QPF xmrg file for the forecast period by determining which of the previous 6 hour synoptic times (i.e. 00, 06, 12,18) has QPF grids available

2. Subtract off any observed precipitation from the QPF

3. Uniformly distribute the remaining QPF into hourly values for the hours remaining in the QPF period

4. Set QPF to zero for any hourly periods beyond the number of hours of requested QPF

Finding the Latest QPF Grid

DHM uses the fact that QPF may be generated at 00, 06, 12, or 18z to locate the most recent QPF grid. In some cases, when QPF is not generated for the most recent 6 hour synoptic time, the grid generated at the next earliest 6 hour synoptic time is used.

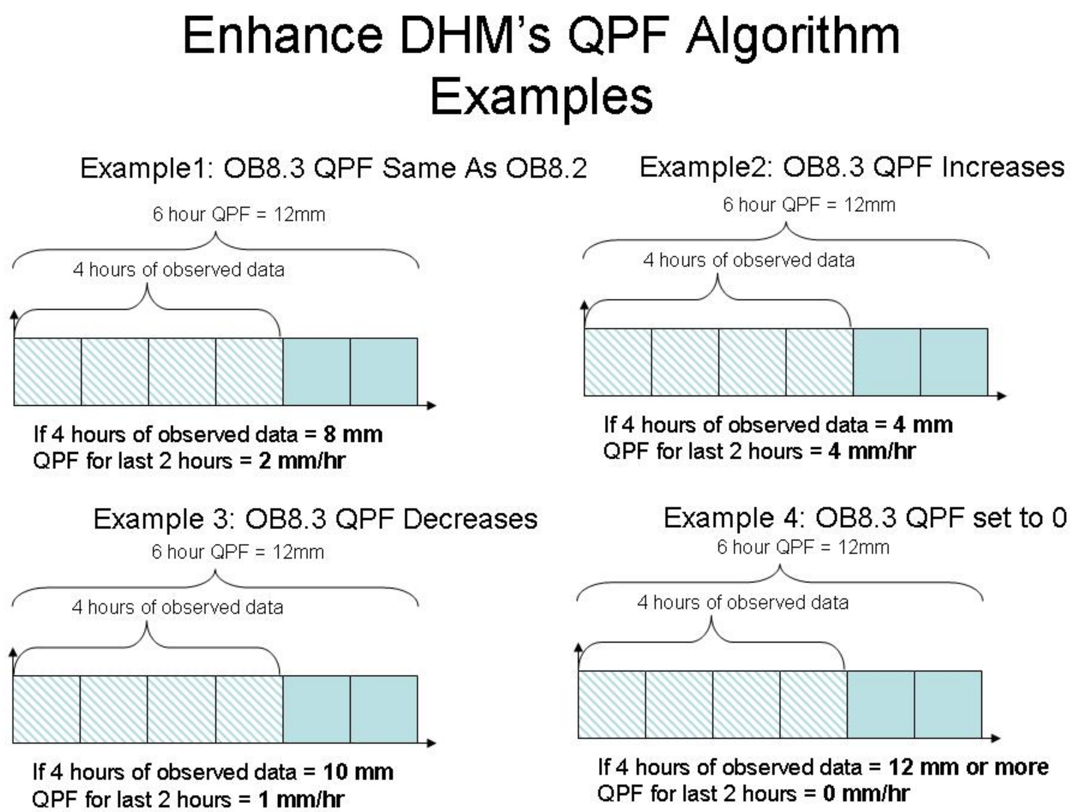
For example for a model run on April 12, 2002 12z, for the forecast period from 12z to 18z, DHM first looks for QPF generated on April 12, 2002 12z. If there are no grids available for 12z, DHM will look for QPF grids generated on April 12, 2002 06z. If no grids are available for 06z, DHM will look for QPF grids generated on April 12 2002 00z, etc.

There is no limit to the number of hours of QPF that can be used by DHM. The number of hours of QPF is set using the FUTPRECP NWSRFS technique. DHM will automatically look for the QPF xmrg file with data for the next 6 hour period. For example if 12 hours of QPF are defined in the segments, for a model run on April 12, 2002 12z, for the forecast period from 18z to 00z, DHM will first look for 2nd period of QPF generated on April 12, 2002 12z. If there are no QPF grids available for 12z, DHM will look for the 3rd period of QPF generated on April 12, 2002 06z. If no grids are available, DHM will look for QPF for the 4th period generated on April 12 2002 00z, etc. For example

files on disk	**Forecast period**	**files containing forecasts for the one hour period**	**File used by DHM**
xmrg6_2003013012f006, xmrg6_2003013012f012, xmrg6_2003013012f018, xmrg6_2003013012f024, xmrg6_2003013006f006, xmrg6_2003013006f012, xmrg6_2003013006f018, xmrg6_2003013006f024, xmrg6_2003013000f018, xmrg6_2003012918f024, xmrg6_2003012912f024	01/30/2003 17z	xmrg6_2003013012f006, xmrg6_2003013006f012, xmrg6_2003013000f018, xmrg6_2003012918f024	xmrg6_2003013012f006

Accounting for Observed Data

The DHM QPF algorithm subtracts off any observed data occurring during the QPF period prior to computing the hourly QPF value. If the remaining QPF is greater than zero, it is equally distributed across the hours remaining in the QPF period. If the remaining QPF is zero or less, QPF for the remaining hours is set to zero. The diagrams below describe the algorithm using examples.



If DHM is run hourly…How it ingests QPF

If the DHM is run on an hourly time step verses a 6 hourly time step, DHM will adjust or slide each hour to include additional QPF. For example the model is run at 15Z, DHM will use QPF from the f006 (contains the 12Z – 18Z QPF), f0012 (contains the 18Z – 00Z QPF), and f018 (contains the 00Z to 06Z QPF). DHM then takes the QPF in each of these 6 hour periods and equally subdivided the QPF into hourly amounts. However Since model is run at 15Z and the XMRG QPE data is available for the 12Z, 13Z and 14Z hours, the models will use the QPE not the QPF for those periods. The QPF period for this example would be 15Z – 02Z.

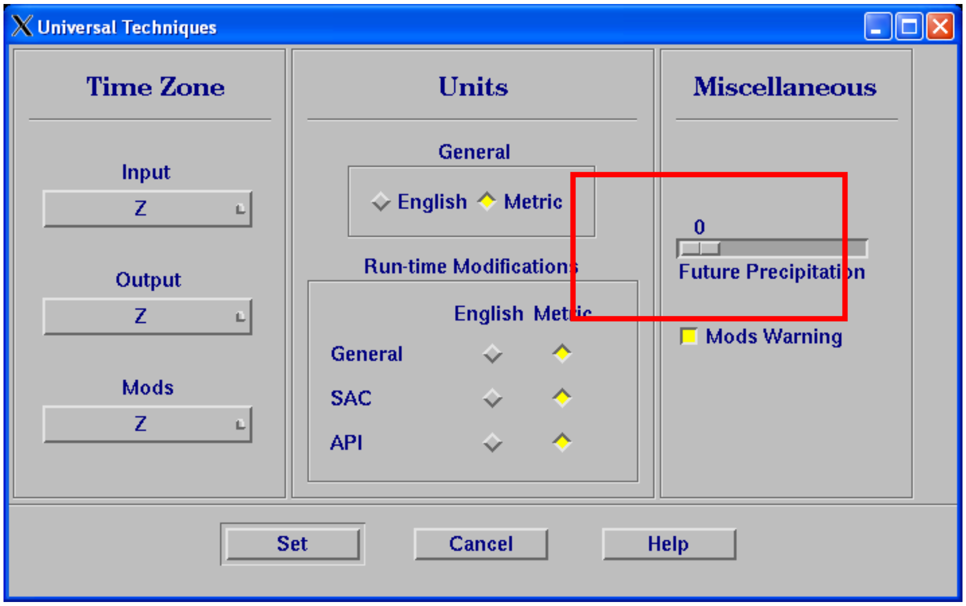
Zeroing out QPF in the IFP GUI

Like existing NWSRFS rainfall-runoff models, DHM uses the FUTPRECP Technique to determine the number of hours of QPF to use VI.5.3D-FUTPRECP : Program FCST HCL Technique FUTPRECP. In batch mode the user defines the number of hours using HCL. For example:

FUTPRECIP (1) = use one hour of QPF
FUTPRECIP (0) = use zero hours of QPF

*** the -1 option, (i.e. FUTPRECIP(-1) = use QPF for all hours is not supported by DHM.

In IFP, the number of hours of QPF is specified using the Universal Techniques GUI.



For all hours beyond the number of hours of requested QPF, the QPF is set to zero for all pixels in the basin.

Updating QPF

Currently, there is no way to update QPF specifically for DHM. If you modify the QPF in NMAP, this will also impact the lumped model simulation since both models use the same QPF input files. This may lead to undesirable results given the temporal, spatial and routing differences between the Lumped Model and DHM. The DPRECIP mod mentioned earlier only edits observed gridded precipitation data, not future precipitation data.

Hence with the functionality of DHM accounting for occurred precipitation (available with the 8.3 build), a scenario such as was highlighted in example 4, may inadvertently cause an undesirable situation. An example would be QPF is forecast in both the first and second 6 hour periods. More rain occurs in the first 6 hour period than initially expected and “all the QPF” is “used up” in the first few hours. Hence the model has zero QPF in the last part of the 6 hour period. However, exists in the second 6 hour period. This may lead to an odd resultant simulation because a gap of “no precipitation” exists even though it is still raining. The second limitation is again other than performing modifications to the hourly QPF xmrg in GFE, the forecaster is not able to make a precipitation mod in IFP to the future precipitation data.