# HEFS Hindcasting Guide

**Version: HEFS 1.2.1**

**Release Date: 28 October 2014**

*National Weather Service*

*Office of Hydrologic Development*

## CONTENTS

# 1. Overview

There are three steps to HEFS hindcasting:

1. The MEFP creates an ensemble of forcings from precipitation and temperature hindcasts.
2. These forcing ensembles are input to an Hydrologic Ensemble Processor (HEP), which generates streamflow ensemble hindcasts.
3. The streamflow hindcasts are bias-corrected with the EnsPost hydrologic post-processor.

The output from MEFP, the hydrologic processor, and EnsPost are exported for verification by the Ensemble Verification System.

This document is a cookbook on how to set up and conduct HEFS streamflow hindcasting in CHPS and export the corresponding results for verification. It assumes that MEFP, the hydrologic processor, and EnsPost have already been configured on an operational Stand-Alone (SA) (see the MEFP Configuration Guide: Forecast Components and the EnsPost Configuration Guide for instructions on how to set up and configure an SA). This document describes how to configure exporting the results of MEFP, the hydrologic processor, and EnsPost, and outlines the steps involved in running the hindcasts.

- As an example, it will configure two operational basins at MARFC, WALN6DEL and CNNN6DEL, in the UpperDelaware forecast group. WALN6DEL is a headwater basin and CNNN6DEL is its downstream basin. You should replace them with two of your own locations in a forecast group, one headwater and one immediately downstream.

This document does not provide scientific guidance on hindcasting, such as the hindcasting scenarios needed to answer particular questions about forecast quality or the necessary length of hindcasts for verification purposes. Examples of hindcasting and verification with the HEFS can be found in the reports that accompany the initial phased validation of the HEFS:

http://www.nws.noaa.gov/oh/hrl/hsmb/hydrologic_ensembles/publications_presentations/index.html

## 1.1 Notation

- Text to be entered as a command line is displayed in `Courier New font`.
- Text in a file name or directory name, text to be entered into an ASCII text file (including xml files), and the file name of any reference document are denoted in *this font*.

## 2. Configuration Changes

This document assumes you run CHPS in an SA and all the configuration files are contained under a single directory (e.g. marfc_sa) referred to <*test_root*> or <*region_dir*>. It assumes that MEFP, HEP streamflow, and EnsPost are already configured in the SA.

You will need to:

1. Prepare MEFP for Hindcasting.
2. Prepare Exports to write the MEFP ensembles, HEP streamflow hindcasts, and EnsPost corrected hindcasts to files.
3. Register these Exports with CHPS.
4. Integrate these Exports into your existing configuration.

In the following example:

- WALN6DEL and CNNN6DEL are our sample locations. Replace them with your sample locations.
- FGroup is WALN6DEL's Forecast Group = UpperDelaware. Replace this with your sample Forecast Group.
- The Export files will be written to $EXPORT_FOLDER$, which is set in your sa.global_properties.

### 2.1 Preparing MEFP for Hindcasting

(Portions of this are excerpted from section 4.1.3 of the MEFP User's Manual.)

You must:

1. Comment out MEFP Preprocessing
2. Add the hindcasting property to MEFP Forecast
3. Select hindcast data
4. Set MEFP_HINDCASTING=true in sa_global.properties (i.e. Turn Hindcasting On)

#### 2.1.1 Comment out MEFP Preprocessing

Unlike the operational SA, the hindcast MEFP does not need to covert gridded data; it will be loaded from the parameter files.

In any Workflow under WorkflowFiles that calls one of the following workflows, XML comment out:

```
<!--  <workflowId>MEFP_Preprocess_RFC_Forecast</workflowId>   -->
<!--  <workflowId>MEFP_Preprocess_GEFS_Forecast</workflowId>  -->
<!--  <workflowId>MEFP_Preprocess_CFSv2_Forecast</workflowId> -->
```

#### 2.1.2 Add Hindcasting Property to MEFP Forecast

In any Module under ModuleConfigFiles that matches:

*_MEFP_TFMN_TFMX_Forecast.xml
*_MEFP_FMAP_Forecast.xml file,

Add the MEFP hindcasting property (**bold** below) to the run_info.xml section:

```
<int key="printDebugInfo" value="0"/>
<string key="hindcasting" value="$MEFP_HINDCASTING$"/>
<string key="parameterDir" value="$MEFP_ROOT_DIR$/mefpParameters"/>
<int key="rfcNumberOfForecastDays" value="0"/>
<int key="gefsNumberOfForecastDays" value="15"/>
<int key="cfsv2NumberOfForecastDays" value="0"/>
<int key="climatologyNumberOfForecastDays" value="0"/>
```

#### 2.1.3 Select Hindcast Data

Set the data you wish to use in the hindcast. For example, in the run_info.xml section,

If you wish to do a 15-day GEFS-only hindcast, set:

```
<int key="rfcNumberOfForecastDays" value="0"/>
<int key="gefsNumberOfForecastDays" value="15"/>
<int key="cfsv2NumberOfForecastDays" value="0"/>
<int key="climatologyNumberOfForecastDays" value="0"/>
```

If you wish to do a 1-year GEFS+CFSv2+Climatology hindcast, set:

```
<int key="rfcNumberOfForecastDays" value="0"/>
<int key="gefsNumberOfForecastDays" value="15"/>
<int key="cfsv2NumberOfForecastDays" value="270"/>
<int key="climatologyNumberOfForecastDays" value="365"/>
```

Hindcasting requires a large and consistent historical sample, ideally 10+ years of data. This may not be available for all forecast sources:

| Source | Used by | Years |
|---|---|---|
| RFC | MEFP | < 10 |
| GEFS | MEFP | 1985-2010 |
| CFSv2 | MEFP | 1982-2010 |
| MAT/MAP | MEFP, EnsPost | varies by RFC |
| SQIN | EnsPost | varies by RFC |
| QME/QINE | EnsPost | varies by RFC |

If your hindcast run will end near the end of a forecast source (2009/2010), you must be careful with the lastEnsembleYear. The base ensemble that MEFP uses as a starting point is constructed from the historical data for each year. In the default configuration, the last ensemble year defined is 2010, which in water years is defined as 10/1/2009 to 9/29/2010:

```
<int key="lastEnsembleYear" value="2010"/>
```

If the last year of your operational run is 2009, and the historical data for a basin ends on 9/29/2009, and you are doing a 15 day GEFS run, not enough historical data will be found after 9/15/2009. Forecasts after 9/15 will generate an error. To test for this error, try a hindcast over September of the last run year. If you see this MEFP error:

```
"Unable to construct base climatology ensemble using historical data;
aborting ensemble generation"
```

reduce the lastEnsembleYear by 1, from 2010 to 2009. The resulting MEFP ensemble will have one fewer member, but forecasts after 9/15 should no longer result in this error.

#### 2.1.4 Turn Hindcasting On

Edit $region_dir/sa_global.properties and add/set:

```
MEFP_HINDCASTING=true
```

To turn hindcasting off, set MEFP_HINDCASTING to false and reverse steps 2.1.1 to 2.1.4.

### 2.2 Export the MEFP Ensembles, HEP Hindcasts, and EnsPost Hindcasts

For each location, create a ModuleConfigFile to export the time series you wish to EVS verify. You should only export required data, as I/O is a significant fraction of the total time required for verification. For example, if you wish to verify over:

- MEFP Precip
- MEFP Temp

- HEP Streamflow
- EnsPost Streamflow

create a file to export the four time series as follows:

- *cd <test_root>/Config/ModuleConfigFiles/hefs/UpperDelaware*
- *nedit WALN6DEL_HEFS_Hindcast_Export.xml*

**Standard Location:** <test_root>/Config/ModuleConfigFiles/hefs/UpperDelaware

**Content of** WALN6DEL_HEFS_Hindcast_Export.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<generalAdapterRun xmlns="http://www.wldelft.nl/fews"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.wldelft.nl/fews
http://chps1/schemas/generalAdapterRun.xsd">
   <general>
      <description>WALN6DEL HEFS Hindcast Export</description>

      <piVersion>1.5</piVersion>

      <rootDir>%REGION_HOME%/Export</rootDir>

      <workDir>%ROOT_DIR%/Export</workDir>

      <exportDir>%REGION_HOME%/Export/hefs</exportDir>

      <exportDataSetDir>REGION_HOME%/Export</exportDataSetDir>
      <exportIdMap>IdExportLAGK</exportIdMap>
      <importDir>%ROOT_DIR%/output</importDir>
      <importIdMap>IdImportLAGK</importIdMap>
      <dumpFileDir>$GA_DUMPFILEDIR$</dumpFileDir>
      <dumpDir>%ROOT_DIR%</dumpDir>
      <diagnosticFile>%ROOT_DIR%/output/diag.fi</diagnosticFile>
      <missVal>-999</missVal>
      <modelTimeStep unit="hour" multiplier="6"/>
   </general>
   <activities>
      <exportActivities>
         <exportTimeSeriesActivity>
            <exportFile>%TIME0%_WALN6DEL_inputs.fi</exportFile>
            <exportBinFile>true</exportBinFile>
            <timeSeriesSets>
               <!-- MEFP precip Export -->

               <timeSeriesSet>
<moduleInstanceId>UpperDelaware_MEFP_FMAP_Forecast</moduleInstanceId>
                  <valueType>scalar</valueType>
                  <parameterId>FMAP</parameterId>
                  <locationId>WALN6DEL</locationId>
                  <timeSeriesType>external forecasting</timeSeriesType>
                  <timeStep unit="hour" multiplier="6"/>
                  <relativeViewPeriod unit="hour" start="0"
startOverrulable="false" end="7920" endOverrulable="false"/>
```

```
                  <readWriteMode>read only</readWriteMode>
                  <ensembleId>MEFP</ensembleId>
                  <ensembleMemberIndexRange start='1961' end='1997'/>
               </timeSeriesSet>

               <!-- MEFP temp Export -->

               <timeSeriesSet>
                  <moduleInstanceId>MEFP_FMAT_Forecast</moduleInstanceId>
                  <valueType>scalar</valueType>
                  <parameterId>FMAT</parameterId>
                  <locationId>WALN6DEL</locationId>
                  <timeSeriesType>external forecasting</timeSeriesType>
                  <timeStep unit="hour" multiplier="6"/>
                  <relativeViewPeriod unit="hour" start="0"
startOverrulable="false" end="7920" endOverrulable="false"/>
                  <readWriteMode>read only</readWriteMode>
                  <ensembleId>MEFP</ensembleId>
                  <ensembleMemberIndexRange start='1961' end='1997'/>
               </timeSeriesSet>

               <!-- HEP Export -->

               <timeSeriesSet>
<moduleInstanceId>ADDSUB_WALN6DEL_ADD.BFQ_Forecast</moduleInstanceId>
                  <valueType>scalar</valueType>
                  <parameterId>SQIN</parameterId>
                  <locationId>WALN6TOT</locationId>
                  <timeSeriesType>simulated forecasting</timeSeriesType>
                  <timeStep unit="hour" multiplier="6"/>
                  <relativeViewPeriod unit="hour" start="0"
startOverrulable="false" end="7920" endOverrulable="false"/>
                  <readWriteMode>read only</readWriteMode>
                  <ensembleId>MEFP</ensembleId>
                  <ensembleMemberIndexRange start='1961' end='1997'/>
               </timeSeriesSet>

               <!-- EnsPost Export -->

               <timeSeriesSet>
                  <moduleInstanceId>WALN6DEL_ENSPOST_Forecast</moduleInstanceId>
                  <valueType>scalar</valueType>
                  <parameterId>SQIN</parameterId>
                  <locationId>WALN6DEL</locationId>
                  <timeSeriesType>simulated forecasting</timeSeriesType>
                  <timeStep unit="hour" multiplier="6"/>
                  <relativeViewPeriod unit="hour" start="0"
startOverrulable="false" end="7920" endOverrulable="false"/>
                  <readWriteMode>read only</readWriteMode>
                  <ensembleId>HEFSENSPOST</ensembleId>
                  <ensembleMemberIndexRange start='1961' end='1997'/>
               </timeSeriesSet>
            </timeSeriesSets>
         </exportTimeSeriesActivity>
      </exportActivities>

      <executeActivities>
```

```
      </executeActivities>

      <importActivities>
      </importActivities>

   </activities>
</generalAdapterRun>
```

This export writes 2 files for the WALN6DEL location, a .fi file (Fast Infoset header) and a .bin file (Binary data). The .fi/.bin pair is compressed and is not human readable. If you wish to export human readable XML, change:

```
<exportFile>%TIME0%_WALN6DEL_inputs.fi</exportFile>
<exportBinFile>true</exportBinFile>
```

to:

```
<exportFile>%TIME0%_WALN6DEL_inputs.xml</exportFile>
<exportBinFile>false</exportBinFile>
```

It is recommended that you export and review a day or two as XML to confirm that the time series are exporting correctly (without missing values).

If you have the disk space, you could export all in XML, but this is not recommended as EVS file import in .fi/.bin format is significantly faster than reading files in XML format. File read times are also sensitive to the number of locations and variables per file. Exporting one location and one variable per file is recommended (e.g. the EnsPost-adjusted streamflow forecasts for WALN6) as more locations and variables per file increases the read time.

This export uses a generalAdapterRun. A previous export technique using a timeSeriesExportRun is slower, and not recommended.

**Note**: EVS discriminates time series based on parameterID and locationID, but not on ensembleId. In the example above, the before/after EnsPost time series are:

```
<parameterId>SQIN</parameterId>
<locationId>WALN6TOT</locationId>
<ensembleId>MEFP</ensembleId>
```

and:

```
<parameterId>SQIN</parameterId>
<locationId>WALN6DEL</locationId>
<ensembleId>HEFSENSPOST</ensembleId>
```

These have different locationIds, so EVS can tell them apart. However, if the HEFSENSPOST locationId happened to be WALN6TOT, the HEFSENSPOST time series would have to be removed from WALN6DEL_HEFS_Hindcast_Export.xml and written by a second export, say WALN6DEL_HEFS_Hindcast_Export2.xml. A future version of EVS will remove this restriction and allow it to also consider ensembleId.

**Note**: The forcing exports should reflect whatever is used for the hydro modeling, so the export should use the timeSeriesSet directly before hydro modeling is done. If you do a MERGETS (Section 2.4.1) in the HEP Preprocessing, replace UpperDelaware_MEFP_FMAP_Forecast above with the merged timeSeriesSet.

**Note:** For hindcasting, EnsPost uses historical observed streamflow, QME (24 hour) or QIN (6 hour), at the same or higher resolution than the EnsPost parameters are calibrated at. If the EnsPost parameters were calibrated at 24 hours, you can use QME or QIN. If the EnsPost parameters were calibrated at 6 hours, you must use QIN.

Add a second location, CNNN6DEL, by creating CNNN6DEL_HEFS_Hindcast_Export.xml.

#### 2.2.1 Create HEFS_Hindcast_Export.xml, a New Workflow

This Workflow calls the 2 modules, WALN6DEL_HEFS_Hindcast_Export.xml and CNNN6DEL_HEFS_Hindcast_Export.xml, created above.

- *cd <test_root>/Config/WorkflowFiles/hefs*
- *nedit HEFS_Hindcast_Export.xml*

**Standard Location:**<test_root>/Config/WorkflowFiles/hefs/

**Content of** HEFS_Hindcast_Export.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<workflow xmlns="http://www.wldelft.nl/fews"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.wldelft.nl/fews
http://chps1/schemas/workflow.xsd" version="1.1">
  <activity>
    <runIndependent>true</runIndependent>
    <moduleInstanceId>WALN6DEL_HEFS_Hindcast_Export</moduleInstanceId>
    <ensemble>
      <ensembleId>MEFP</ensembleId>
      <runInLoop>false</runInLoop>
    </ensemble>
  </activity>
  <activity>
    <runIndependent>true</runIndependent>
    <moduleInstanceId>CNNN6DEL_HEFS_Hindcast_Export</moduleInstanceId>
    <ensemble>
      <ensembleId>MEFP</ensembleId>
      <runInLoop>false</runInLoop>
    </ensemble>
  </activity>
</workflow>
```

#### 2.2.2 Register Export Modules and Workflow with CHPS

- *cd <test_root>/Config/RegionConfigFiles/*

- *nedit ModuleInstanceDescriptors.xml*
- Add the following lines to the bottom of the file, before </moduleInstanceDescriptors>:

```
<moduleInstanceDescriptor id="WALN6DEL_HEFS_Hindcast_Export">
          <moduleId>GeneralAdapter</moduleId>
</moduleInstanceDescriptor>
<moduleInstanceDescriptor id="CNNN6DEL_HEFS_Hindcast_Export">
          <moduleId>GeneralAdapter</moduleId>
</moduleInstanceDescriptor>
```

- *cd <test_root>/Config/RegionConfigFiles/*
- *nedit WorkflowDescriptors.xml*
- Add the following lines to the top of the file, after <workflowDescriptors>:

```
<workflowDescriptor id="HEFS_Hindcast_Export" name="HEFS Hindcast Export"
forecast="false" visible="false" allowApprove="false" autoApprove="false">
           <description>HEFS Hindcast Export</description>
</workflowDescriptor>
```

## 2.3 Integrating Exports with Your Existing Configuration

You can either add the export to an existing workflow, or create (copy) a new workflow just for hindcasting. In this example we create a new workflow. It doesn't matter, as long as the following tasks are completed by some workflow:

MEFP Precip Ensemble Generation
MEFP Temp Ensemble Generation
HEP Preprocessing
HEP Streamflow
EnsPost Streamflow
(new) Hindcast Exports

The exports must be done at the end of each T0; you can't run all the T0s and then export - this would require all hindcasts to be retained in the localDataStore.

### 2.3.1 Create a New Hindcasting Workflow

- *cd <test_root>/Config/WorkflowFiles/hefs*
- *nedit HEFS_Hindcast.xml*

Add the following content to the file opened:

**Standard Location:**<test_root>/Config/WorkflowFiles/hefs/

**Content of** HEFS_ Hindcast.xml

```
<workflow xmlns="http://www.wldelft.nl/fews"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.wldelft.nl/fews
http://chps1/schemas/workflow.xsd" version="1.1">

<!-- Already configured, Preprocessing is turned off -->
  <activity>
    <runIndependent>false</runIndependent>
    <workflowId>MEFP_Forecast</workflowId>
  </activity>

<!-- Already configured, and includes HEFS preprocessing -->
  <activity>
    <runIndependent>false</runIndependent>
    <workflowId>UpperDelaware_HEFS_Forecast</workflowId>
  </activity>

<!-- EnsPost Forecast is done in UpperDelaware_HEFS_Forecast -->

<!-- New for Hindcasting -->
  <activity>
    <runIndependent>false</runIndependent>
    <workflowId>HEFS_Hindcast_Export</workflowId>
  </activity>
</workflow>
```

#### 2.3.2 Register Hindcasting Workflow with CHPS

- *cd <test_root>/Config/RegionConfigFiles/*
- *nedit WorkflowDescriptors.xml*
- Add the following lines to the top of the file, after <workflowDescriptors>:

```
<workflowDescriptor id="HEFS_Hindcast" name="HEFS Hindcast" forecast="true"
visible="true" allowApprove="true" autoApprove="true">
   <description>HEFS Hindcast</description>
    <cardinalTimeStep id="12Z"/>
      <stateSelection>
         <warmState>
            <stateSearchPeriod start="-6" end="0" unit="day"/>
         </warmState>
      </stateSelection>
      <runExpiryTime unit="hour" multiplier="1"/>
</workflowDescriptor>
```

When copying or editing an operational workflow there are 2 changes to make:

- Set the end day of the warmState stateSearchPeriod to zero (bold above). The workflow will use the most recent warm states.

- Set the runExpiryTime to 1 hour to limit the localDataStore size and delete the data produced by HEFS_Hindcast after 1 hour (bold above). To take advantage of this, CHPS must be run with the rolling barrel on (fews_ohdPlugins.sh).

When the rolling barrel is on, the runExpiryTime is activated and all expired records in the localDataStore will be deleted. This may accidentally delete the warm states. To avoid this, either:

a. Always recreate your warm states right before the hindcast, or

b. Add a long runExpiryTime :

```
<runExpiryTime unit="year" multiplier="1"/>
```

to the workflow that generates the warm states (see 3.1.4). Make sure the warm states are generated with that setting. That way they will not expire.

To run CHPS with the rolling barrel on, run fews_ohdPlugins.sh. Warm states shouldn't expire, hindcasts will expire, and your localDataStore will not grow large.

To run CHPS with the rolling barrel off, run fews_ohdPlugins.sh.rboff. Warm states won't expire, hindcasts won't expire, but your localDataStore may grow large.

## 2.4 HEP Modification

The Hydrologic Ensemble Processor may need to modify the input data.

### 2.4.1 Merging Precip Time Series

SNOW17 may use MERGETS to merge two sources of precip data, MAP and MAPX, into a single time series. If MAP is input first, but the MAP is not found, it will fill in 0 values and not be over-written by MAPX. The warm states needed in section 3.1 will be incorrect or useless.

Assume LL (ll) is your location. If you have a file like:

```
$configuration_dir/ModuleConfigFiles/ll/MERGETS_LL_MAP_UpdateStates.xml
```

then MERGETS merges MAP and MAPX data. If you cannot find this file, then you are using only a single source of precip, and need to do nothing. Otherwise:

- *cd <test_root>/Config/ModuleConfigFiles/ll/*
- *nedit MERGETS_LL_MAP_UpdateStates.xml*

In MERGETS_LL_MAP_UpdateStates.xml, the merge looks like:

```
<merge>
```

```
  <simple>
    <inputVariable>
       <variableId>LL_MAP_6</variableId>
    </inputVariable>
    <inputVariable>
       <variableId>LL_MAPX_6</variableId>
    </inputVariable>
    <moduleParameterFile>MERGETS_LL_MAP_UpdateStates</moduleParameterFile>
    <outputVariable>
        <variableId>LL_MAPX_6</variableId>
    </outputVariable>
  </simple>
</merge>
```

XML comment out the MAP:

```
<!--
    <inputVariable>
       <variableId>LL_MAP_6</variableId>
    </inputVariable>
-->
```

Make the same change in the MERGETS Forecast module.

- *cd <test_root>/Config/ModuleConfigFiles/ll/*
- *nedit MERGETS_LL_MAP_Forecast.xml*

Find the <merge> area, and XML comment out the MAP:

```
<!--
    <inputVariable>
       <variableId>LL_MAP_6</variableId>
    </inputVariable>
-->
```

You will have to do this for each Location (LL/ll) in your hindcast.

MERGETS_LL_MAP_Forecast is run during your HEP preprocessing.

#### 2.4.2 MAPE Time Series

SAC-SMA uses the MAPE time series. If <FG> (UpperDelaware) is your Forecast Group, open:

```
$configuration_dir/ModuleConfigFiles/preprocessing_esp/FG_SampleESP_Forecast.
xml
```

And look for the MAPE timeSeriesSet:

```
      <variableId>MAPE_ESP</variableId>
```

If you cannot find it, then MAPE is hard-coded into your SAC-SMA and you need to do nothing.

Otherwise, MAPE must be produced for the hindcast. Sample ESP can be used. It is produced by running the pre-processing workflow, UpperDelaware_PreProcessing_Forecast.xml. This workflow should be updated by adding the module instance to produce MAPE ensembles:

- *cd <test_root>/Config/WorkflowFiles/UpperDelaware/*
- *nedit UpperDelaware_PreProcessing_Forecast.xml*

In the file opened, add the content:

```
<activity>
      <runIndependent>true</runIndependent>
      <moduleInstanceId>SetTimes_ESP_Forecast</moduleInstanceId>
</activity>
<activity>
      <runIndependent>true</runIndependent>
      <moduleInstanceId>UpperDelaware_SampleESP_Forecast</moduleInstanceId>
</activity>
```

- *cd <test_root>/Config/ModuleConfigFiles/preprocessing_esp/*
- *nedit UpperDelaware_SampleESP_Forecast.xml*

Change:

```
<variable>
      <variableId>MAPE_Historic</variableId>
      <timeSeriesSet>
            <moduleInstanceId>ImportSHEF</moduleInstanceId>
            <valueType>scalar</valueType>
            <parameterId>MAPE</parameterId>
            <locationSetId>Meteo_UpperDelaware</locationSetId>
            <timeSeriesType>external historical</timeSeriesType>
            <timeStep id="12Z"/>
            <relativeViewPeriod unit="day" start="-36500" end="0"/>
            <readWriteMode>read only</readWriteMode>
      </timeSeriesSet>
</variable>
```

to:

```
<variable>
      <variableId>MAPE_Historic</variableId>
      <timeSeriesSet>
            <moduleInstanceId>ImportSHEF</moduleInstanceId>
            <valueType>scalar</valueType>
            <parameterId>MAPE</parameterId>
            <locationSetId>Meteo_UpperDelaware</locationSetId>
            <timeSeriesType>external historical</timeSeriesType>
            <timeStep id="12Z"/>
```

```
            <relativeViewPeriod unit="day" start="-36500" end="0"/>
            <readWriteMode>read complete forecast</readWriteMode>
      </timeSeriesSet>
</variable>
```

The readWriteMode change is necessary to for UpperDelaware_SampleESP_Forecast to produce different MAPE traces.

Hindcasting uses only MEFP ensembles, so you must change the ensembleId inside the SampleESP MAPE timeSeriesSet to pretend it came from MEFP:

- *cd <test_root>/Config/ModuleConfigFiles/preprocessing_esp/*
- *nedit FG_SampleESP_Forecast.xml*

Change:

```
<ensembleId>ESP</ensembleId>
```

to:

```
<ensembleId>MEFP</ensembleId>
```

You will have to do this for each Forecast Group (FG) in your hindcast.

FG_SampleESP_Forecast is run during your HEP preprocessing.

## 3. Example Run

Hindcasting must be run with the hydrologic models in a warm state.

### 3.1 Produce the Warm States

#### 3.1.1 Import Historical Data

To produce the warm states for the hydrologic models, run the models with historical observed forcings. The forcings must be in the database. To import the forcings, run the workflow ‘Import Scalars’ (or your RFC equivalent).

‘Import Scalars’ imports MAP, MAT, and QME. If MAPE is input for the rainfall-runoff model (SAC-SMA and API), it should also be imported. For MARFC, MAPE data is also imported by ‘Import Scalars’.

Historical data is required on a 12Z forecasting clock of {18Z,12Z,0Z,6Z}. For historical data recorded in a local time system other than CST (UTC-6), a time shift is needed to map the data to the 12Z forecasting clock. In practice, the closest time system for all RFCs is UTC-6 or CST. If your RFC does **not** use CST,

- *cd <test_root>/Config/ModuleConfigFiles/import/*
- *nedit ImportDataCard.xml*

Change:

```
<timeZoneOffset>+00:00</timeZoneOffset>
```

to:

```
<timeZoneOffset>-06:00</timeZoneOffset>
```

wherever you find it in the file.

### 3.1.2 Run UpdateStates Workflow

After importing the historical data, run a workflow to produce warm states. Your RFC should already have a workflow configured to do this. Our example uses the workflow 'UpperDelaware UpdateStates'.

Select 'UpperDelaware UpdateStates' in the Manual Forecast Display, Select the three modules which cover the locations we will run, as shown in Figure 3.

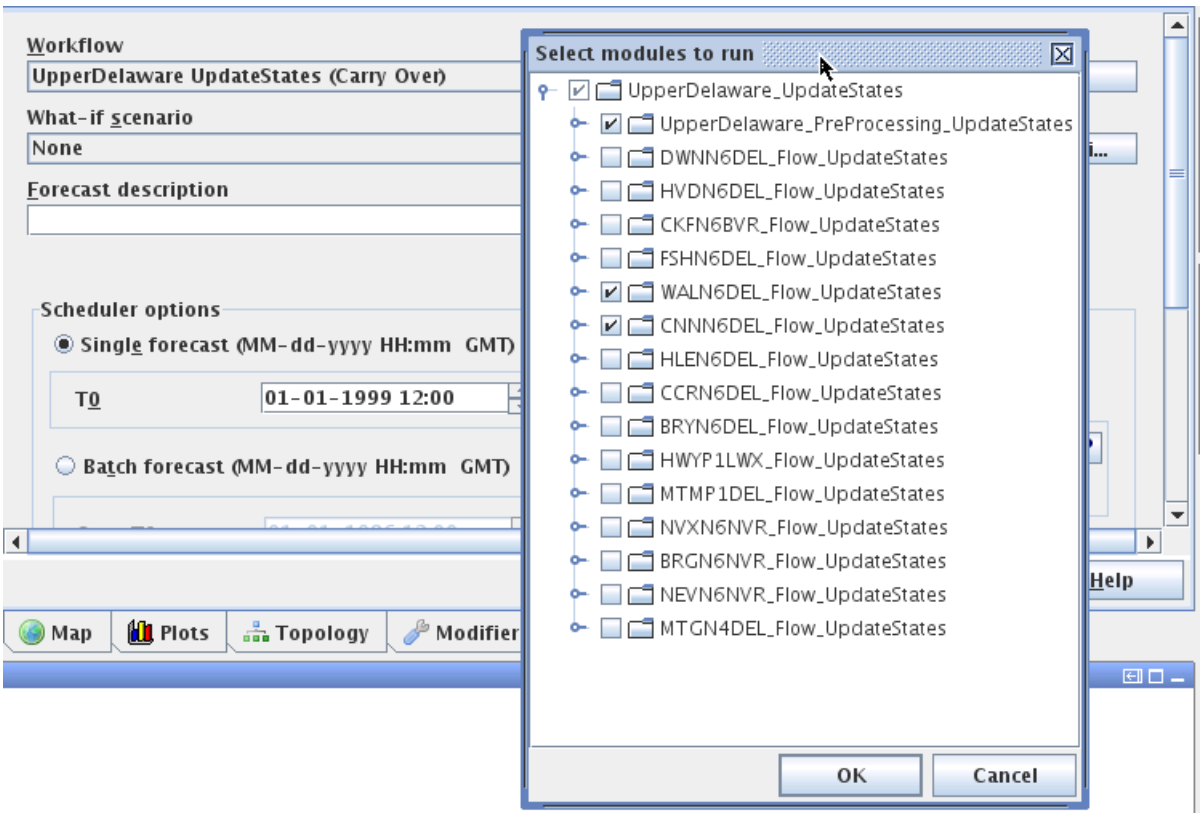


Figure 3. Select appropriate module instances to run

Assume we intend to generate hindcasts over 20 years, from 01-01-1979 to 01-01-1999.

An appropriate warm-up period, longer than one year, is picked to allow the models to arrive at a warm state at the start of the hindcasting period. This is known as the “spin-up” time. We will use a 2-year spin-up period, from 01-01-1977 to 01-01-1979.

Run a hydrologic simulation from the start of the spin-up period through to the end of the hindcast period as a ‘Single’ forecast ‘as in Figure 4.

Select initial state in “State selection” to start in a cold state at the beginning of our warm up period (hindcast start – 2 years).

Check “Approve” so that the model states from this run are saved in the database for hindcast use. There is no need to change the forecast length.

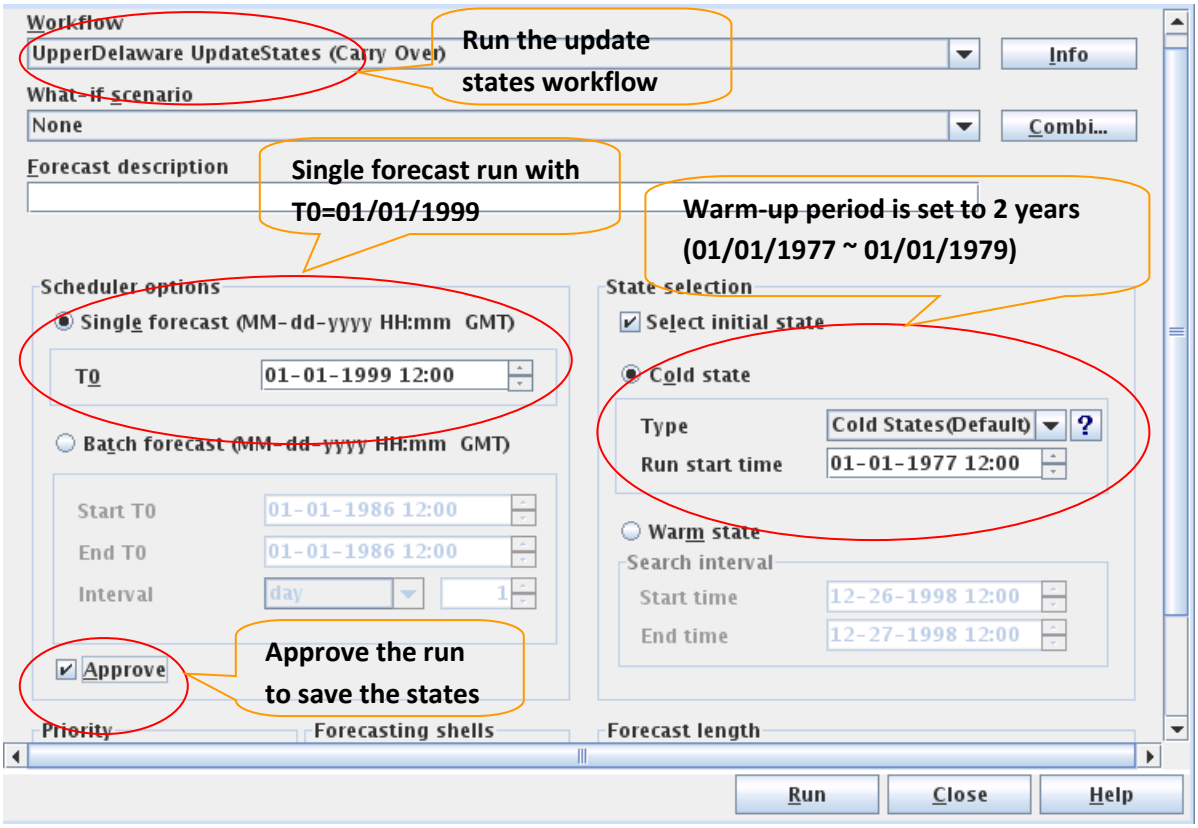


Figure 4. Update states run to generate warm states

### 3.2 Generate Hindcasts

Running the hindcast workflow, HEFS_Hindcast, will produce MEFP ensembles, HEP streamflow hindcasts, and EnsPost post-processed hindcasts.

The forecast interval and forecast length should match the data you are using to generate the MEFP ensembles (set in section 2.1.3). In this example we are doing a 330-day GEFS+CFSv2+Climatology hindcast. CFSv2 has some leap day considerations; see section 3.3 for a further discussion.

Start CHPS with fews_ohdPlugins.sh.rboff (do not delete the warm states in the database):

```
% cd $work_dir

% ./hefsPlugins/fews_hefsPlugins.sh.rboff marfc_sa &
```

In the CHPS Manual Forecast display (refer to Figure 5):

- Select the workflow "HEFS_ Hindcast"
- Check "Batch forecast" in Scheduler options; specify Start T0 to be "01-01-1986 12:00:00", End T0 to be "01-05-1986 12:00:00", with "1 day" as the interval.
- Define "330 days" as the forecast length.
- Press "Run"
- During the run, the .xml data will be written to the $EXPORT_FOLDER$ directory, which is <test_root>/Export/

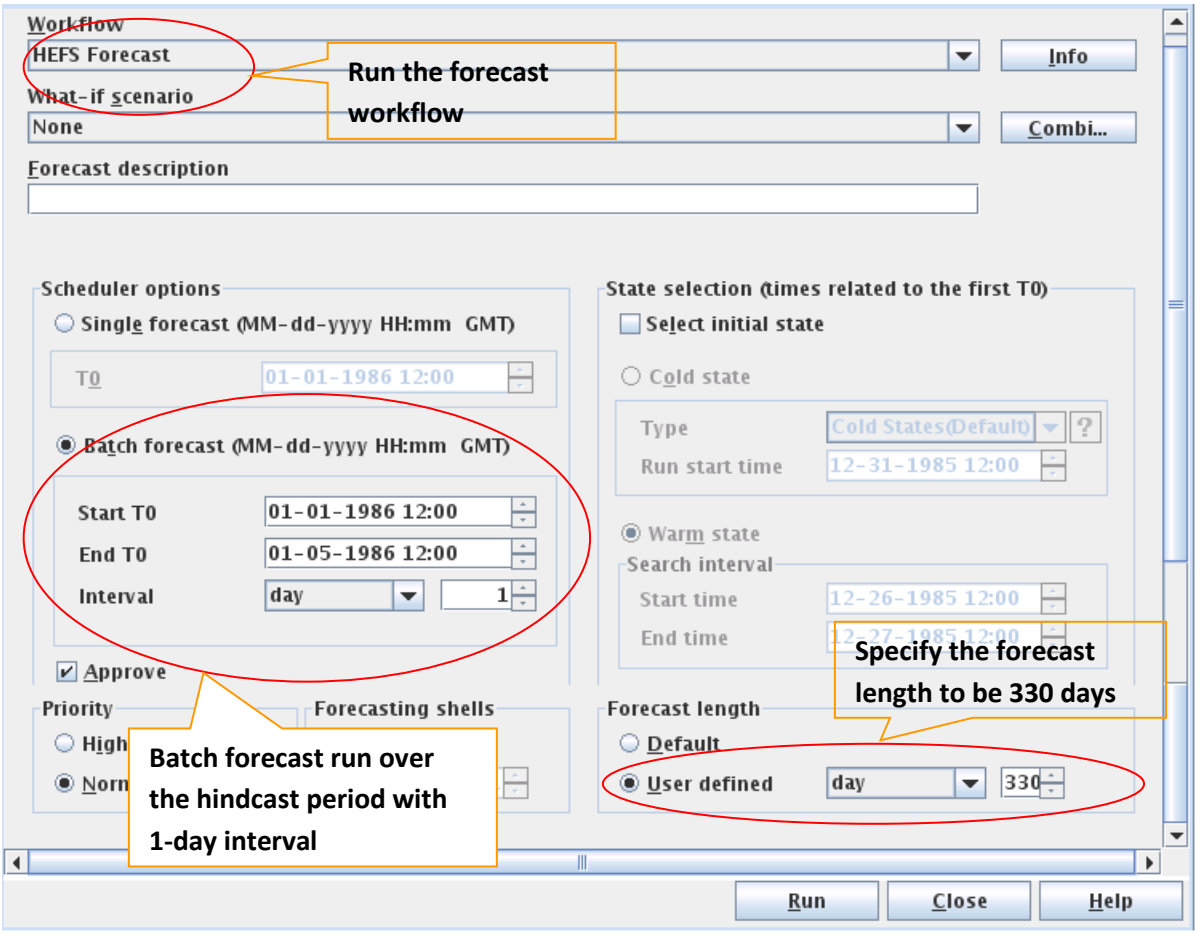


Figure 5. Run the forecast workflow in a batch mode to generate the hindcasts

The MEFP ensembles, HEP streamflow, and EnsPost-processed streamflow can be viewed in the CHPS Database Viewer:

Inside the Logs Panel of CHPS display:

➢ Press “F12”, and then select “J”. This will open the CHPS Database Viewer.

a) Select a T0 that you are interested in viewing (e.g., 01-05-1986 12:00:00; refer to Figure 6a) All of the time series corresponding to this T0 will show up.

b) Select a module instance of interest (e.g., UpperDelaware_MergeMAPX_Forecast and UpperDelaware_MergeMAT_Forecast which produce MEFP hindcasts for the hydrologic processor); select the time series corresponding to these two module instances for ensemble ID 1961, as shown in Figure 6a.

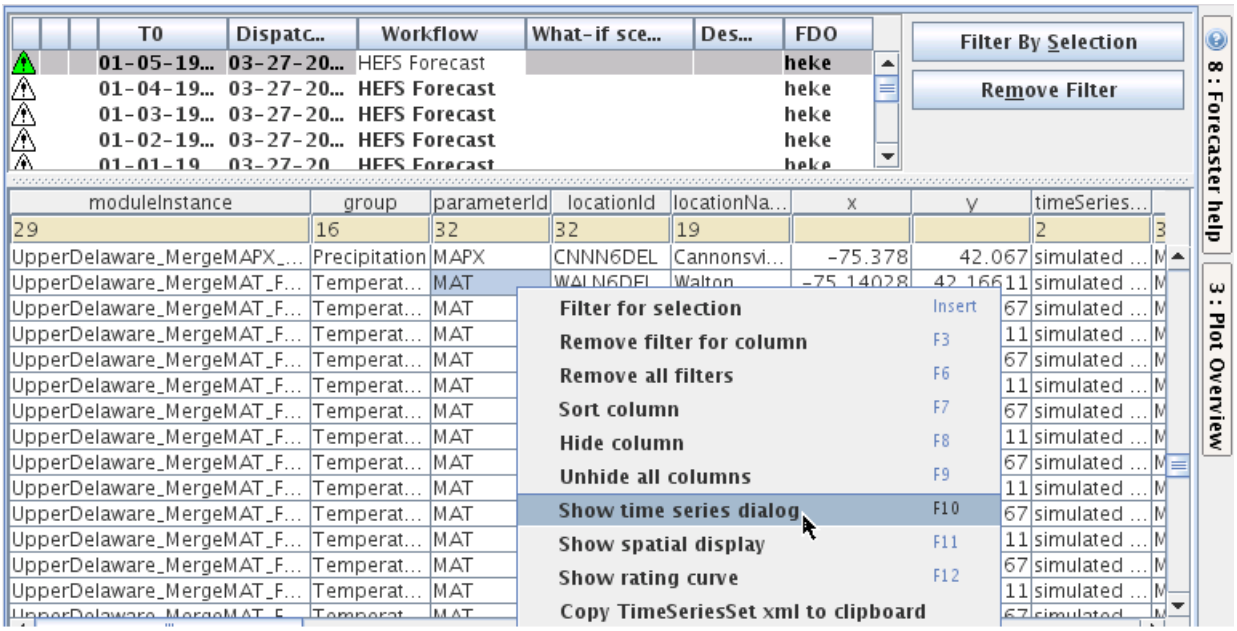


Figure 6a: Select time series to plot

c) Once the time series of interest has been selected, right click and select Show time series dialog F10 (refer to Figure 6a). The selected time series will be plotted, as illustrated in Figure 6b.

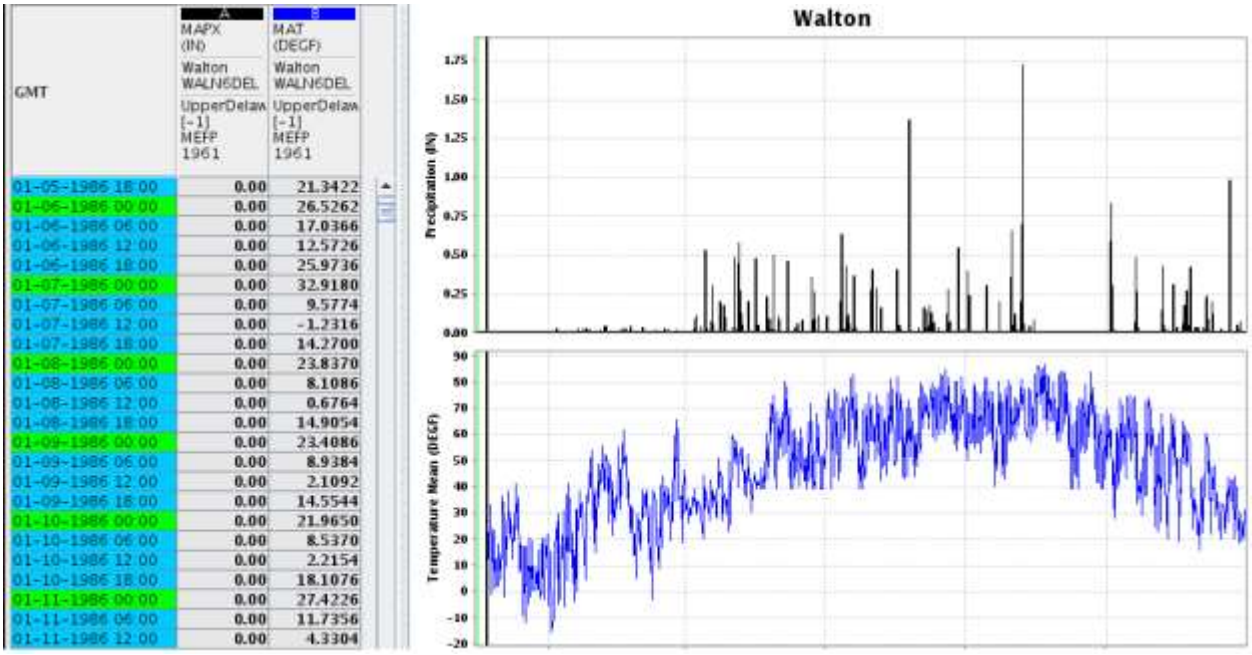


Figure 6b. MEFP hindcasts for WALN6DEL (Ensemble ID: 1961)

d) Repeat steps b) and c) for other module instances of interest (e.g., ADD.SUB_WALN6DEL_ADD.BFQ_Forecast for streamflow hindcasts produced

for WALN6DEL (Figure 6c); WALN6DEL_ENSPOST_Forecast, for EnsPost-processed hindcasts for WALN6DEL (Figure 6d).

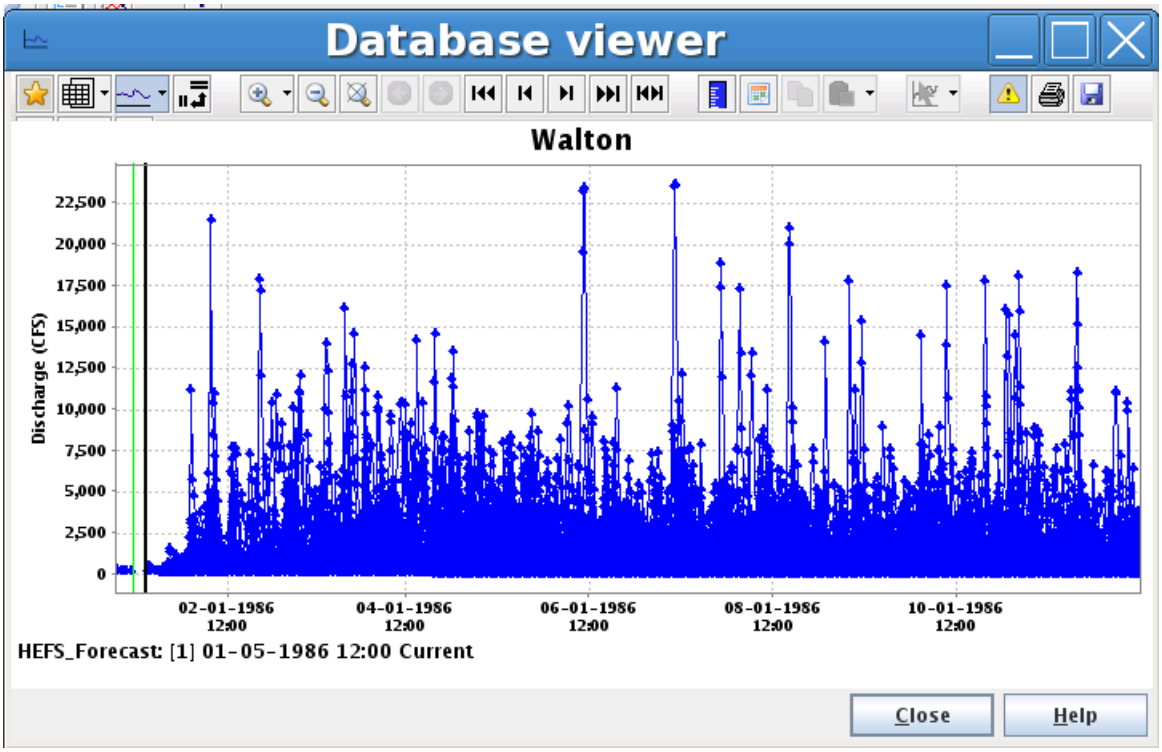


Figure 6c: Plot of streamflow hindcasts at WALN66DEL

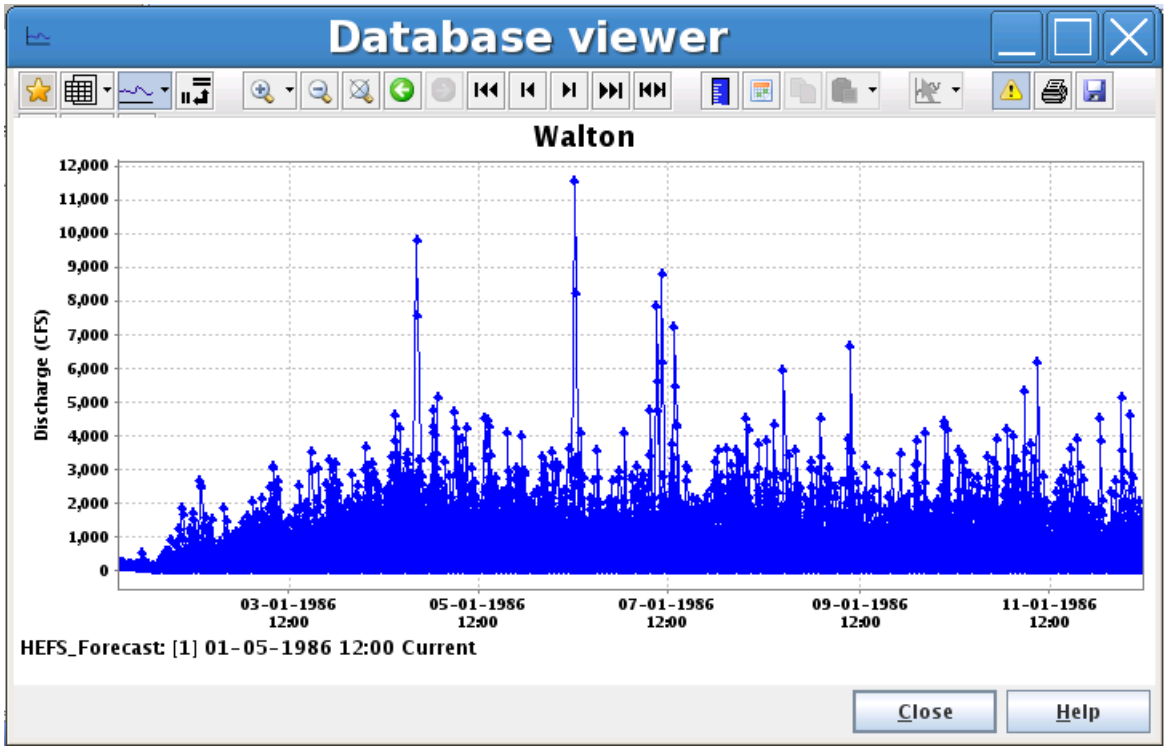


Figure 6d: EnsPost-processed streamflow hindcasts at WALN66DEL.

e) Repeat steps a) through d) for other T0s of interest.

### 3.3 CFSv2 and Leap Days

If you are not using CFSv2 as a data source (section 2.1.3), there are no leap-day considerations.

As a non-CFSv2 example, assume you wish to do a GEFS-only hindcast from 1985-01-15 to 1995-01-15 with a 15-day forecast horizon. In the Manual Forecast window you would set the Forecast Interval to 1 day, Forecast Length to User defined 15 days, Start T0 = 1985-01-15, End T0 = 1995-01-15, and hit the Run button.

If you are using CFSv2, you should be aware that CFSv2 reforecasts are available on a 5-day cycle that begins on January 1 and ignores leap days. To get consistent results when the hindcasting period crosses leap days, a series of 5-day, leap-day avoiding hindcasts is recommended.

The CHPS batch forecaster schedules, then runs forecasts, so you don't have to wait for one forecast batch to complete before scheduling the next one.

As a CFSv2 example, assume you wish to do a GEFS+CFSv2+Climatology hindcast (see section 2.1.3) from 1985-01-01 to 1998-12-31 with a one year forecast horizon. In

the Manual Forecast window you would set the Forecast Interval to 5 days and the Forecast Length to User defined 1 year. Schedule the hindcasts as follows to avoid leap days:

Start T0 = 1985-01-01, End T0 =1988-**02-28**, hit Run button.
Start T0 = 1988-**03-02**, End T0 =1992-02-28, hit Run button.
Start T0 = 1992-03-02, End T0 =1996-02-28, hit Run button.
Start T0 = 1996-03-02, End T0 =1998-12-31, hit Run button.

When all the hindcasts are scheduled, you can minimize CHPS and return periodically to check the run progress.

### 3.4 Run EVS Workflow

The EVS workflow can be run to produce verification products.

Select workflow “EVS_Test_Workflow” in Manual Forecast display

Check “Single forecast” in Scheduler options

Leave anything else unchecked in the display

Press “Run”.

EVS products will be exported to /Models/ohd/evs/output/

Use the “display” command to view the output. For example:

```
% cd <test_root>/Models/ohd/evs/output/
% display BLKO2.QINE.HMOS.Correlation_coefficient.jpg
```

## 4. Tips and Troubleshooting

### 4.1 Terminating a Hindcast

In the Logs panel, select F12 to open the Database Viewer; then select M = Kill running forecast.

### 4.2 Turn Off the Screen Saver

During hindcasts, the screen saver may disrupt the display and stop the hindcast run. To avoid this potential problem, disable the screen saver as follows:

a. Left click on the Linux Desktop, and select Configure Desktop.
b. Select Display, then Power Control.

c. Keep Enable Display Power Management checked.
d. Note default values on all 3 sliders.
e. Select Disable on all 3 sliders.
f. Click OK to save.

To re-enable the screen saver, return the 3 sliders to their default values.

### 4.3 Check Hindcast is Exporting

In the discussion below,

$region_dir is your <*test_root*> directory.

$export_root_dir is your EXPORT_ROOT directory, $region_dir/Export

Instead of opening the Database viewer (section 3.2), you can also inspect the .xml files in your EXPORT_ROOT directory:

```
% cd $export_root_dir/EnsPost
% ls -lrt
```

The timestamps will give you an estimate of how long each T0 is taking. To check that the files are being filled, here is a bash script to count the non-missing events (missVal is set to -999 in section 2.2):

```
for i in *
do
   echo
   echo -n "$i not missing:"
   grep event $i | grep -v 999 | wc -l
   echo -n "$i     missing:"
   grep event $i | grep    999 | wc -l
done
```

This script will only work if the exportFile in section 2.2 is set to .xml instead of .fi/.bin. If events are missing, something may be wrong with the hindcast.

### 4.4 Check All T0s Were Run

The previous script only looks inside export T0s that were run. Here is another bash script, check_exports.sh, to check for the T0s you scheduled to run. The arguments are:

```
% check_exports.sh <start_t0> <end_t0> <days> <files_per_t0>
```

where:

```
<start_t0>          is the T0 to start checking
<end_t0>            is the T0 to stop checking
<days>              is the number of days between T0s
```

```
   <files_per_interval> is the number of files you expect per interval
```

For example, to check a hindcast that starts at 19850115, ends at 19950115, with 1 day between T0s, which should generate 2 files per T0, use:

```
% check_exports.sh 19850115 19950115 1 2
```

Before running the script, set EXPORT_FOLDER to the directory you want to check exports, WORKFLOWID to the name of the Workflow you ran, and FORECAST_LENGTH to your hindcast length.

```
#!/bin/bash
# Script to check for missing T0 in exports
# check_exports.sh <start T0> <end T0> <# of interval days> <#number of
expected exports per T0>
# Example: $./check_exports.sh 19850101 19950101 1 56

START_DATE=$1
END_DATE=$2
INTERVAL=$3
EXPECTED_NUMBER=$4

EXPORT_FOLDER=/nobk/ensembles1/wardj/cnrfc_sa/Export/EnsPost

#batch XML created for missing T0 to be run by macro
REFORECAST_XML=reforecast.xml
WORKFLOWID=HEFS_Hindcast
FORECAST_LENGTH=15
INIT_XML=0

#batch xml initiation function
initialize(){
        echo '<?xml version="1.0" encoding="UTF-8"?>' > $REFORECAST_XML
        echo '<taskPropertiesPredefined xmlns="http://www.wldelft.nl/fews"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.wldelft.nl/fews
http://fews.wldelft.nl/schemas/version1.0/taskPropertiesPredefined.xsd">' >>
$REFORECAST_XML
   echo '' >> $REFORECAST_XML
}

#add task to batch xml
add_task(){
   echo '   <taskProperties>' >> $REFORECAST_XML
   echo '      <description>reforecast batch</description>' >>
$REFORECAST_XML
   echo '      <workflowId>'$WORKFLOWID'</workflowId>' >> $REFORECAST_XML
   echo '      <taskSelection>' >> $REFORECAST_XML
   echo '         <singleTask>' >> $REFORECAST_XML
   echo '            <time0>'$(date -d "$CUR_DATE" +%Y-%m-
%d)'T12:00:00.000+00:00</time0>' >> $REFORECAST_XML
   echo '         </singleTask>' >> $REFORECAST_XML
   echo '      </taskSelection>' >> $REFORECAST_XML
   echo '      <forecastLength unit="day" multiplier="'$FORECAST_LENGTH'"/>'
>> $REFORECAST_XML
```

```
   echo '      <runExpiryTime unit="hour" multiplier="1"/>' >>
$REFORECAST_XML
   echo '   </taskProperties>' >> $REFORECAST_XML
   echo '' >> $REFORECAST_XML
}

#loop to iterate through each T0
for (( CUR_DATE=$START_DATE; "$CUR_DATE" <= "$END_DATE"; CUR_DATE=$(date -d
"$CUR_DATE +$INTERVAL day" +%Y%m%d)))
do
   #verify number of expected exports
   FILES=$(ls $EXPORT_FOLDER/$CUR_DATE* | wc -l)
   if [ $FILES != $EXPECTED_NUMBER ]
   then
      echo "Export does not match for T0 $CUR_DATE"
      if [ $INIT_XML = 0 ]
      then
         initialize
         add_task
         INIT_XML=1
      else
         add_task
      fi
   fi
done
#close the xml tag if file is created
if [ $INIT_XML = 1 ]
then
   echo '</taskPropertiesPredefined>' >> $REFORECAST_XML
   echo $REFORECAST_XML" was created for missing T0s"
else
   echo 'No missing exports'
fi
```

**Note**: check_exports.sh reports leap days as missing T0s. It is assumed if you hindcast around leap years the start and end dates you provide for the script will also avoid them.

### 4.5 Re-running Missing T0s

It is recommended that you complete your hindcast by re-running any missing T0s. Back up your log.txt if you wish to investigate why a T0 failed. A re-run will overwrite log.txt.

Re-running can be done as single forecasts (one T0 at a time), or by using a FEWS Macro. To enable FEWS Macros, add the lines to your $region_dir/Config/DisplayConfigFiles/ManualForecastDisplay.xml the top of the file, after <manualForecastDisplay>:

```
<runningPredefined>
<description>Run pre-defined forecasts</description>
<directory>%REGION_HOME%</directory>
<buttonVisible>true</buttonVisible>
</runningPredefined>
```

On the next run a Macro button will be visible next to your Run button.

check_exports.sh produces a file called reforecast.xml that is a FEWS Macro. Move this file into your REGION_HOME directory. For example, if you are missing two T0s of 1991-06-15 and 1991-06-15, reforecast.xml will look like.

```
<?xml version="1.0" encoding="UTF-8"?>
<taskPropertiesPredefined xmlns="http://www.wldelft.nl/fews"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.wldelft.nl/fews
http://fews.wldelft.nl/schemas/version1.0/taskPropertiesPredefined.xsd">

        <taskProperties>
                <description>reforecast batch</description>
                <workflowId>HEFS_Hindcast</workflowId>
                <taskSelection>
                        <singleTask>
                                <time0>1991-06-15T12:00:00.000+00:00</time0>
                        </singleTask>
                </taskSelection>
                <forecastLength unit="day" multiplier="15"/>
                <runExpiryTime unit="hour" multiplier="1"/>
        </taskProperties>

        <taskProperties>
                <description>reforecast batch</description>
                <workflowId>HEFS_Hindcast</workflowId>
                <taskSelection>
                        <singleTask>
                                <time0>1993-11-15T12:00:00.000+00:00</time0>
                        </singleTask>
                </taskSelection>
                <forecastLength unit="day" multiplier="15"/>
                <runExpiryTime unit="hour" multiplier="1"/>
        </taskProperties>
</taskPropertiesPredefined>
```

Restart CHPS, and go to Manual Forecast. The Macro button will appear next to the Run button. Click on the Macro button and select reforecast.xml to run the Macro.

To rule out possible localDataStore (CHPS database) corruption before re-running, you can refresh your localDataStore directory with one containing only the warm states.

See also the Deltares documentation:

http://publicwiki.deltares.nl/display/FEWSDOC/Add+Macro+Button#AddMacroButton-Relatedmodulesanddocumentation

### 4.6 Investigate Why a T0 Failed

To investigate why a T0 failed, look at log.txt

When a CHPS batch forecast is run, the T0s are scheduled (dispatched), and then they are run. If a T0 is missing, it is either because it was not scheduled, or (more likely) it was scheduled and the run failed due to a data error.

Assume we are missing a T0 of 1991-06-15. To check if it was scheduled, look in the log for a line contains 1991-06-15 and text like:

“INFO - ManualForecastDialog.dispatchTasks - ManualForecast.Started: Manual forecast”. To do this, perform the following commands:

```
% cd $region_dir
% grep -nr '06-15-1991' log.txt
```

Dispatching is done early in the batch and the grep should return quickly. To check whether a scheduled task ran, look for the T0 in log.txt for text like:

“TaskRun.Started: Starting Task HEFS_Hindcast with ID 2366 and T0 1991-06-15 12:00:00 and forecast length of 15 days”

log.txt can be a large file (7 Gigabytes) and it can be difficult to search in the middle of it. To work around this, cut 10,000 lines after the task start, and write it to a second file. The following commands will perform this:

```
% grep -A10000 'and T0 1991-06-15 12:00:00 and forecast length' log.txt >
1991-06-15.txt
% vi 1991-06-15.txt
```

This T0 logged about 7000 lines; the extra 3000 were to guarantee we overlapped with the next T0.

Look in 1991-06-15.txt for ERROR. This T0 hit a timeout of 5 minutes, possibly due to a high machine load:

```
“[2014-01-27 01:33:14,728] ERROR - WorkflowPluginActivity$1.call -
ohd.hseb.ohdfewsadapter.OHDFewsAdapter failed with error Execution of Java
class ohd.hseb.ohdfewsadapter.OHDFewsAdapter terminated after time out of 300
seconds while running ExecuteJavaActivity: /awips/chps/ohd/3.3.a/bin
ohd.hseb.ohdfewsadapter.OHDFewsAdapter "arguments:
/tmp/FEWS_CNRFC_SNOW17_WBGC1H_WBGC1HUF_Forecast_MEFP_1977_9062953735510577241
/run_info.fi " for ensemble MEFP_1977
nl.wldelft.fews.system.plugin.generaladapter.GeneralAdapterException:
Execution of Java class ohd.hseb.ohdfewsadapter.OHDFewsAdapter terminated
after time out of 300 seconds version: NWS2013.01, build: 43108 Nov7, type:
stable, jre: 1.7.0_11, mx: 954m, db: Firebird, region: cnrfc_sa+”
```

### 4.7 Speeding up Hindcasts

If you have more than one processor core available, the temperature part of the MEFP run can be parallelized using the FEWS Multi-core option (FogBugz issue 1150). To count the number of available processors:

```
% cat /proc/cpuinfo
```

And look for the last lines like:

processor : 3
vendor_id : GenuineIntel

This machine has 4 processors (0-3). To utilize all 4 processors:

a. In your MEFP_Forecast workflow (the workflow that calls MEFP_FMAT_Forecast), run the ensembles in a loop:

```
<activity>
    <runIndependent>true</runIndependent>
    <moduleInstanceId>MEFP_FMAT_Forecast</moduleInstanceId>
    <ensemble>
        <ensembleId>MEFP</ensembleId>
        <runInLoop>true</runInLoop>
    </ensemble>
</activity>
```

b. In your MEFP_FMAT_Forecast module, XML comment out all of the ensembleId lines:

```
<!-- <ensembleId>MEFP</ensembleId> -->
```

c. If you are running a single SA on your machine, add the following line to your sa_global.properties:

```
runInLoopParallelProcessorCount=100
```

Even though this machine has only 4 processors, Deltares recommends setting it to 100 to use all of them (the max available). If you are running multiple SAs on your machine, set the following in each of their sa_global.properties:

```
runInLoopParallelProcessorCount=2
```

This directs each SA to use no more than 2 processors, one for CHPS, and one for the system. The SAs will not compete for processors. It is also recommended that each SA have a separate temporary working directory and a separate piServicePortNumber , so your sa_global.properties for SA 1 would add:

```
tempDir=/tmp/hindcast_1
piServicePortNumber=8101
runInLoopParallelProcessorCount=2
```

and your sa_global.properties for SA 2 would add:

```
tempDir=/tmp/hindcast_2
piServicePortNumber=8102
runInLoopParallelProcessorCount=2
```

You will have to create the tempDirs. If you do not give them separate temporary directories, SA 2 may wait for SA 1 to finish writing (and vice versa), and you can get timeouts, causing missing T0s. For run speed, the temporary directories should be on a disk local to where the SA is running (usually under /tmp).

To count the number of processors being used while a hindcasting is running:

```
% ps -e -o psr= | sort | uniq | wc -l
```

returns:

```
% 4
```

### 4.8 Running Out of Disk Space

One file, log.txt (CHPS log), and 2 directories, the localDataStore (CHPS database) and the Export directory (where the .xml is written) will grow during your hindcast. For example, after 10 years of a daily (29 MEFP/14 HEP/2 EnsPost) location hindcast:

- log.txt increased by 7 Gigabytes
- localDataStore increased by 38 Gigabytes
- Export increased by 71 Gigabytes.

To monitor their sizes, use du –sh:

```
% du -sh $region_dir/log.txt
% du -sh $region_dir/localDataStore
% du –sh $export_root_dir
```

To reduce the size of log.txt, edit Log4jConfig.xml in your SA directory. In the root section at the bottom of the file:

```
<root>
     <priority value="INFO"/>
     <appender-ref ref="defaultLogFile"/>
     <appender-ref ref="splashScreen"/>
     <appender-ref ref="explorerLogPanel"/>
     <appender-ref ref="explorerAcknowladgeIcon"/>
```

```
    <appender-ref ref="dataStoreLogEntriesTable"/>
</root>
```

You can change the priority value to the following:

| Priority value | Will write |
|---|---|
| ERROR | only ERROR |
| WARN | WARN and ERROR |
| INFO (default) | model output, INFO, WARN, and ERROR |

Without INFO messages it can be difficult to tell if a run has started or ended. You can use 'top' to see if FEWS is running and 'ls –lrt' check the Export file times.

If you have additional disk space away from the SA directory, you can move your localDataStore and Export directory and link to it before running (log.txt cannot be moved):

```
% cd $region_dir
% ln -s /big_disk/localDataStore localDataStore
% ln -s /big_disk/Export Export
```

However, you may pay an I/O penalty in run speed if the disk is not local to where the SA is running.

The large Export directory size was due to exports in XML, using .fi/.bin compression should reduce it by a factor of 10.

### 4.9 Increase Java Memory

The maximum size of your Java memory allocation pool (-Xmx) is set in the script you start CHPS. For example, ./hefsPlugins/fews_hefsPlugins.sh.rboff uses the settings:

```
... -XX:MaxPermSize=128m -Xmx1024M $JVM_OPTS ...
```

You can increase it as follows:

```
... -XX:MaxPermSize=128m -Xmx2048M $JVM_OPTS ...
```

The Xmx value should be a multiple of 1024M. You can also run the JVM as a server instead of a client (the default). Add -server:

```
... -XX:MaxPermSize=128m -Xmx2048M -server $JVM_OPTS ...
```

You can get a startup boost by setting the minimum size of your Java memory allocation pool (-Xms) to 1024M:

```
... -XX:MaxPermSize=128m -Xms1024M -Xmx2048M -server $JVM_OPTS ...
```

Your SA will start with 1 Gig of memory already allocated.

To check the settings were applied:

```
% ps -ef | grep java
```

Should return something like:

```
% .../jre/bin/java -XX:MaxPermSize=128m –Xms1024M -Xmx2048M –server …
```

This may be useful only if you are running 1 SA, as multiple SAs will wait for memory to become available before launching.

### 4.10 Keep Hindcast Configuration Current

If you see a log.txt error message like:

"[2013-12-18 19:09:13,667] ERROR - ExecuteActivity.processDiagnosticFile - GA.Execution.Model.Error: In adapter model: LAGKMODEL LocationID(CBNK1): java.lang.Exception: Error: could not set execute permission to file : /awips/chps_share/ohd/bin/lagk Check that file exist or if application can write it.",

this indicates that the hindcasting configuration is not up to date with the operational configuration. Your RFC should have updated to use the Java version of lagk and the hindcasting configuration is looking for a FORTRAN version of lagk, which is no longer provided after CORE-CHPS-3.3.a, in April 2014.

The hindcasting configuration should be updated with all operational configuration changes, especially changes to baseline CHPS.

### 4.11 Using the Hindcast Robot

You can use a hindcast robot to automate hindcasts. The steps are detailed in the Hindcast Robot Manual. In summary, they are:

- Install the robot software.
- Add the location of the robot (fews_hindcast_robot.jar) and the robot libraries (sikuli_libs) to your PATH.
- Edit a config file (config.txt below) to tell the robot the Start T0/End T0 you want to run (this can span several years). You also tell it where your Export (output) directory is as the robot checks for and re-run skipped T0s.
- The robot controls the mouse, so we recommend you run inside a vncserver/vncviewer pair to give the robot a virtual desktop to run in.

- Start the robot and wait for it to complete. You will run a command like:

```
../jre/bin/java -jar fews_hindcast_robot.jar ./config.txt
```

If you use the Hindcast Robot, the Hindcasting Workflow must be at the top of the WorkflowDescriptors.xml.

The forecastLength must also be set:

- *cd <test_root>/Config/DisplayConfigFiles/*
- *nedit ManualForecastDisplay.xml*
- Add the following lines to the bottom of the file, before </manualForecastDisplay>:

```
<forecastLength unit="day" multiplier="330"/>
```

If you run multiple robots, we recommend using a maximum of 1/2 the available CPUs on a machine as SAs. Each SA needs system CPU time to write /tmp directories and allocate Java threads. The system load (uptime) increased by 2 for every SA we ran, so 3 SAs = an additional load of 6. The higher load didn't seem to affect the T0 runtimes too much. The first SA took about 2 minutes per T0 = 12 hours per hindcast year. The last SA took about 3 minutes per T0 = 18 hours per hindcast year.

We also recommend each robot run in its own vncserver instead of multiple robots sharing a vncserver. The vncservers don't consume much system resources.

## *5.* Configuration Checklist

| | Action | See Section |
|---|---|---|
| | Comment out MEFP Preprocessing | 2.1.1 |
| | Add hindcasting property to MEFP Forecast | 2.1.2 |
| | Select hindcast MEFP data | 2.1.3 |
| | Turn on hindcasting in sa_global.properties | 2.1.4 |
| | Create HEFS export modules | 2.2.1 |
| | Register HEFS export modules | 2.2.2 |
| | Create Hindcasting workflow | 2.3.1 |
| | Register Hindcasting workflow | 2.3.2 |
| | Merging Precip Time Series | 2.4.1 |
| | MAPE Time Series | 2.4.2 |
| | | |

## 6. Run Checklist

| | Action | See Section |
|---|---|---|
| | Import Historical Data | 3.1.1 |
| | Update Warm States | 3.1.2 |
| | Forecast length matches MEFP data | 3.2 |
| | CFSv2 skips leap days | 3.3 |
| | Screen saver disabled | 4.2 |
| | Hindcast exports.fi/.bin files | 4.3 |
| | All T0s were run | 4.4 |
| | Missing T0s re-run | 4.5 |
| | MEFP temp uses multiple cores | 4.7 |
| | Disk space is OK | 4.8 |

## References

EnsPost Configuration Guide, HEFS 0.3.2 Release, OHD, April 2013

Hindcast Robot Manual, OHD, March 21, 2014

MEFP Configuration Guide: Forecast Components, HEFS 0.3.2 Release, OHD, April 2013

MEFP User's Manual, HEFS 0.3.2 Release, OHD, April 2013

Verification of Medium-range Forecasts from the NWS Hydrologic Ensemble Forecast Service (HEFS) with Forcing Inputs from the Frozen Version of NCEP's Global Forecast System (GFS), Technical Report, OHD, March 2013.