

Guidance on Conducting Streamflow Hindcasting in CHPS

Release #

Release Date

**National Weather Service
Office of Hydrologic Development**

I. Document history

Action	Person	Date
First draft	kevin.he@noaa.gov	16 th November 2011
Updated draft	kevin.he@noaa.gov	1 st December 2011
Input from Lee Cajina	Lee.Cajina@noaa.gov	10 th January 2012
Added Appendix by Limin	Limin.Wu@noaa.gov	19 th January 2012
Edits and additions	kevin.he@noaa.gov	June 22 nd 2012
Input from James Brown	james.brown@hydrosolved.com	June 25 th 2012
Input from Limin Wu	Limin.Wu@noaa.gov	June 28 th 2012
Version 2 created	kevin.he@noaa.gov	July 3 rd 2012
Input from Nick Heen	Nickolas.heen@noaa.gov	July 5 th 2012
Version 3 created	kevin.he@noaa.gov	July 9 th 2012

CONTENTS

1.	Overview	4
1.1	Notation.....	4
2.	Configuration	5
2.1	Prerequisite	5
2.2	General Configuration.....	5
2.3	Configuration of the Hydrologic Processor.....	7
2.3.1	Create new location set for the test locations	8
2.3.2	Create/update carryover workflow and relevant configuration files	9
2.3.3	Create/update forecast workflow and relevant configuration files.....	11
2.4	Configuration of EnsPost.....	20
2.4.1	Create EnsPost forecast module instances	21
2.4.2	Create EnsPost export workflow and module instances.....	25
2.4.3	Add EnsPost forecast and export files to the parent workflow	27
3.	Example Run	27
3.1	Produce Initial Model States	28
3.1.1	Model spin-up-creating initial states for the start of the hindcast run.....	28
3.1.2	Generate initial states for each hindcast time	29
3.2	Generate Streamflow Hindcasts and EnsPost-processed Data	30
4.	Appendix: Generation of MEFP Hindcasts Outside of CHPS	34
5.	References	38

1. Overview

Streamflow hindcasting with the HEFS differs from the regular ESP hindcasting conducted by the RFCs. In particular, the MEFP is calibrated with precipitation and temperature hindcasts from several Numerical Weather Prediction (NWP) models, as well as climatological forcing. These forcing ensembles are input to a hydrologic processor, which generates the streamflow ensemble hindcasts. The streamflow hindcasts are then bias-corrected with a hydrologic post-processor, EnsPost.

HEFS hindcasting configuration differs from HEFS operational configuration in the MEFP forcings fed to the hydrologic processor. In an operational run, MEFP is run within CHPS to produce ensemble forcing (precipitation and temperature) forecasts corresponding to the forecast time (T0). The forcing data are in grib format. In hindcasting, MEFP is run outside of CHPS (for now). It produces ensemble forcing hindcasts for multiple hindcast times (e.g., for every day in a 20-year period). The forcing data is in datacard format. As such, in hindcasting extra configuration is required to import forcing hindcasts for every hindcast issue time.

Using two operational basins at ABRFC (CBNK1 and BLKO2 in the KEYINF forecast group, where CBNK1 is a headwater basin and BLKO2 is its downstream basin) as an example, this document describes the configuration of the hydrologic processor and the EnsPost. It also outlines the steps involved in running the hindcasts with the hydrologic processor and EnsPost. Accordingly, this guidance consists of two parts: the configuration of the hydrologic processor and EnsPost, and an example run.

It is worth noting that this document serves as a “cook book” on how to set up and conduct HEFS streamflow hindcasting in CHPS. This document will not provide scientific guidance on hindcasting, such as the necessary length of hindcasts for verification purposes. For more information on the scientific guidance of hindcasting, the reader is referred to a companion document, A Strategy for Science Evaluation of the Hydrologic Ensemble Forecast Service (HEFSv1) (OHD, 2012).

1.1 Notation

- Text to be entered as a command line is displayed in *italics*.
- Text in a file name or directory name, and text to be entered into an ASCII text file (including xml files) is denoted in this font.

2. Configuration

There are four stages of configuration need to be completed: prerequisite, preliminary configuration, configuration of the hydrologic processor, and configuration of the EnsPost. This section describes these necessary configurations in details.

2.1 Prerequisite

This guidance assumes you will run CHPS in a stand-alone mode and all the configuration files are contained in a single directory (e.g., `abrfc_sa.hindcast`) referred to as `<test_root>` hereinafter.

The configuration is predicated upon having HEFS 0.1.2 installed. Particularly, the HEFS bin directory should be defined in the file `sa_global.properties`. The following instructions show how to add the HEFS bin directory:

- `cd <test_root>`
- `nedit sa_global.properties`
- add a line “HEFSBINDIR=\$MODELDIR\$/hefs/bin” in the content (with quotation mark removed)

2.2 General Configuration

This guide further assumes the MEFP hindcasts have been produced outside of CHPS in advance. Detailed descriptions on running MEFP ensemble generators to produce precipitation and temperature hindcasts can be found in the MEFP User’s Manual (OHD, 2012). The Appendix also provides three example scripts to illustrate the running of the MEFP in stand-alone mode (not within CHPS) to generate the required forcing hindcasts. In the hindcasting run, these data will be copied to the import directory (of CHPS) for ingest by CHPS to drive the hydrologic models and produce streamflow hindcasts sequentially (for each issue time and ensemble trace). This first requires creating two new directories, a data directory that contains the MEFP hindcasts and an import directory, where the MEFP hindcasts will be imported to CHPS. It is also necessary to create a shell script to copy the MEFP hindcasts (corresponding to a specified issue time) from the data directory to the import directory. This can be achieved by following the instructions below:

- `cd <test_root>/Import`
- `mkdir copy_cardfiles_hindcast_MEFP`
- `mkdir cardfiles_hindcast_MEFP`

where directory `copy_cardfiles_hindcast_MEFP` is the data directory (storing MEFP hindcasts) and the directory `cardfiles_hindcast_MEFP` is the import directory. The import directory needs to be added to the file `sa_global.properties`:

- `cd ..`
- `nedit sa_global.properties`
- add a line `"IMPORT_FOLDER_DATACARD_HINDCAST=$IMPORT_FOLDER_ROOT$/cardfiles_hindcast_MEFP"` in the content (with quotation mark removed)

Note “*nedit*” is used as the xml file editor throughout this document. The readers are encouraged to use any other editor they prefer in their tests. Now create a file named Copyfiles (to copy MEFP hindcasts from the data directory to the import directory):

- `cd <test_root>/Import`
- `nedit Copyfiles`

In the file opened, add the content in the table below:

Standard Location: `<test_root>/Import/`

Content of Copyfiles
<pre>#!/bin/ksh fromDir=\$1 toDir=\$2 Mask=\$3 echo "Origin directory \$1" >> [your directory]/loginfo.txt echo "Destination directory \$2" >> [your directory]/loginfo.txt echo "Mask \$3" >> [your directory]/loginfo.txt cp \$1/\$3BLKO2.MAT \$2/ cp \$1/\$3BLKO2.MAP06 \$2/ cp \$1/\$3CBNK1.MAT \$2/ cp \$1/\$3CBNK1.MAP06 \$2/</pre>

After saving and closing the file, make it executable:

- `chmod 744 Copyfiles`

In addition to creating the directories for MEFP hindcasts, a new directory named “hefs” should also be created in various locations to store HEFS-related configuration files:

- `cd <test_root>/Config/ModuleConfigFiles/`
- `mkdir hefs`
- `cd <test_root>/Config/WorkflowFiles/`
- `mkdir hefs`
- `cd <test_root>/Config/ModuleDataSetFiles/`
- `mkdir hefs`

The following diagram outlines these directories and the shell script file created, as well as their proper locations in the `<test_root>` file structure.

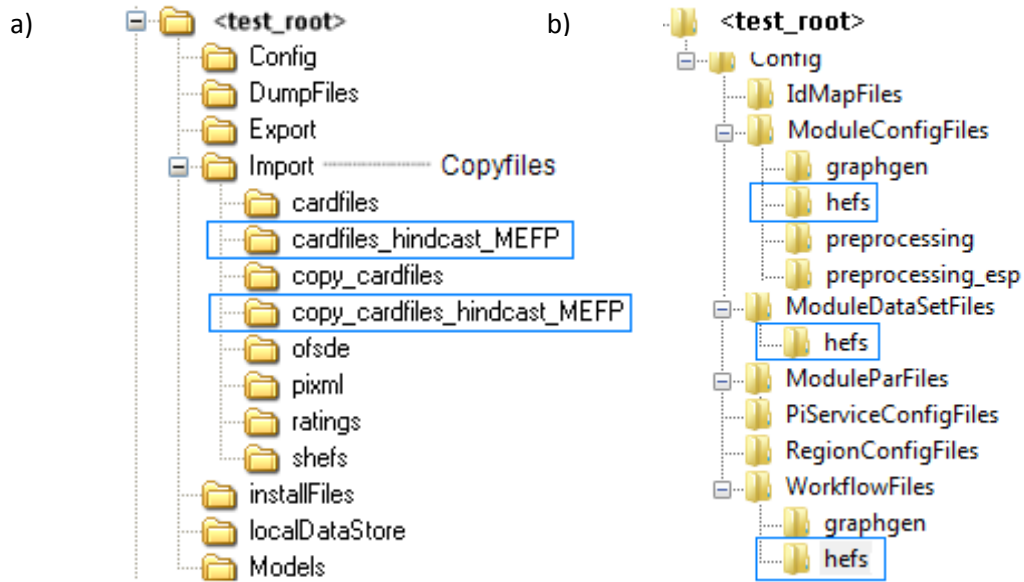


Figure 1. File structure for general configuration: a) file structure for MEFP hindcasts; b) file structure for hefs directories.

It is worth noting that the example files Copyfiles and sa_global.properties (and other files mentioned in the rest of this document) are provided along with this guidance.

2.3 Configuration of the Hydrologic Processor

Configuring the hydrologic processor involves configuring the carryover workflow (and its associated configuration files to generate initial states) and the forecast workflow (and its associated configuration files to digest MEFP hindcasts and produce streamflow hindcasts) for the test locations. Figure 2 depicts the (xml) files (in addition to the module files for hydrologic models themselves) that must be created and/or modified, as well as their proper locations in the <test_root> file structure. Those files are also described below in detail and provided along with this guidance.

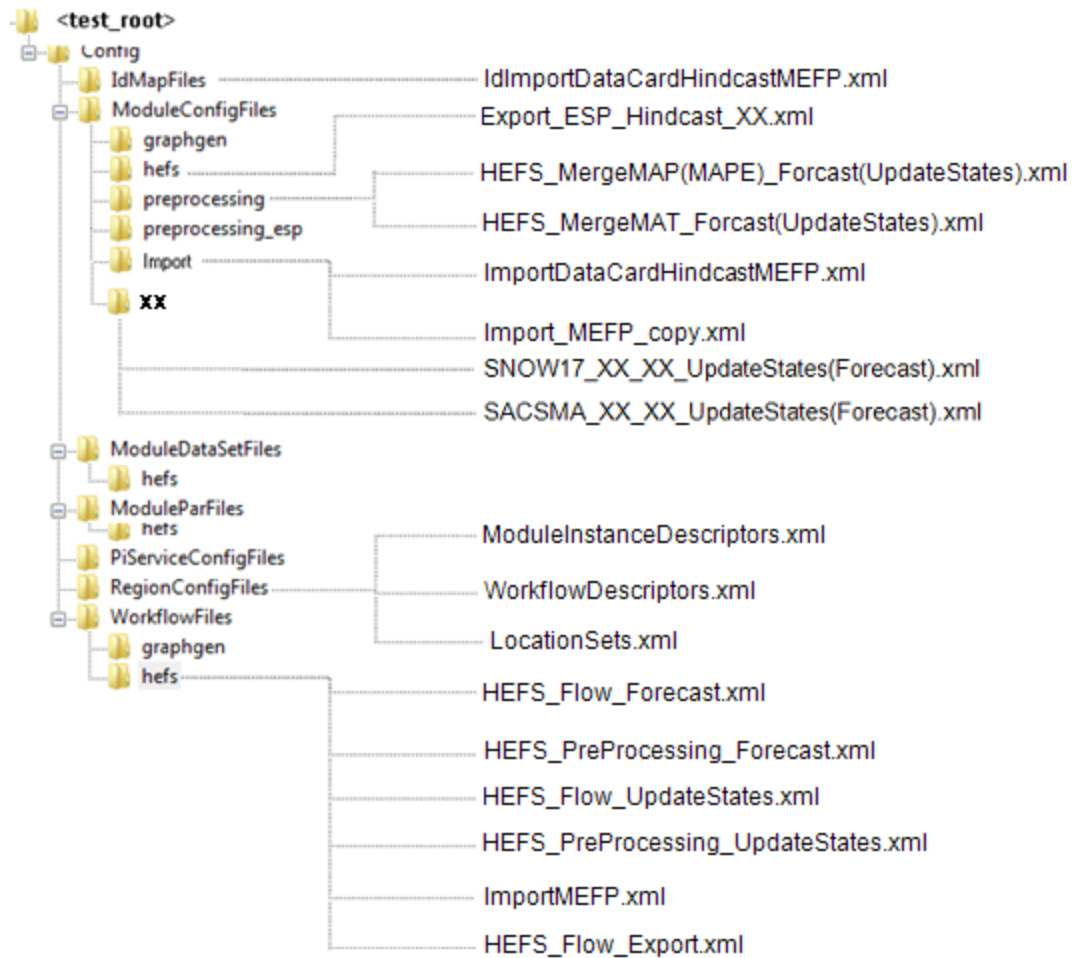


Figure 2. File structure for configuration of the hydrologic processor

2.3.1 Create new location set for the test locations

This is to create a location set exclusively containing the test locations CBNK1 and BLKO2 by modifying the file LocationSets.xml. This location set will be applied in subsequent configuration.

- *cd <test_root>/Config/RegionConfigFiles/*
- *nedit LocationSets.xml*
- add the following part to the file


```

      <locationSet id="Catchments_HEFS">
        <locationId>CBNK1</locationId>
        <locationId>BLKO2</locationId>
      </locationSet>
      
```


2.3.2 Create/update carryover workflow and relevant configuration files

2.3.2.1 Create carryover workflow

This is to create a carryover workflow HEFS_Flow_UpdateStates.xml and a corresponding preprocessing workflow HEFS_PreProcessing_UpdateStates.xml. The latter executes three merge modules to merge the historical data with the MEFP hindcasts. The merged data serves as input for the hydrologic models. The carryover workflow calls the preprocessing workflow and then runs the hydrologic models for the test locations to produce model states.

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

In the file opened, add the content in the table below:

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

Contents: HEFS_Flow_UpdateStates.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>false</runIndependent>
    <workflowId>HEFS_PreProcessing_UpdateStates</workflowId>
  </activity>
  <activity>
    <runIndependent>false</runIndependent>
    <workflowId>CBNK1_Flow_UpdateStates</workflowId>
  </activity>
  <activity>
    <runIndependent>false</runIndependent>
    <workflowId>BLKO2_Flow_UpdateStates</workflowId>
  </activity>
</workflow>
```

- *nedit HEFS_PreProcessing_UpdateStates.xml*

In the file opened, add the content in the table below:

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

Content of HEFS_PreProcessing_UpdateStates.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>false</runIndependent>
    <moduleInstanceId>HEFS_MergeMAT_UpdateStates</moduleInstanceId>
  </activity>
  <activity>
    <runIndependent>false</runIndependent>
    <moduleInstanceId>HEFS_MergeMAP_UpdateStates</moduleInstanceId>
  </activity>
  <activity>
    <runIndependent>false</runIndependent>
    <moduleInstanceId>HEFS_MergeMAPE_UpdateStates</moduleInstanceId>
  </activity>
</workflow>
```

Now register the workflow files just created in WorkflowDescriptors.xml:

- *cd <test_root>/Config/RegionConfigFiles/*
- *nedit WorkflowDescriptors.xml*
- add the following content to the file:

```
<workflowDescriptor id="HEFS_PreProcessing_UpdateStates" name="HEFS_PreProcessing_UpdateStates" forecast="true"
visible="false ">
```

2.3.2.2 Create merge module instances

This step configures the three merge modules files called by HEFS_PreProcessing_UpdateStates.xml (refer to 2.3.2.1). It should be noted that for ABRFC, MAPE data is required as an input to the hydrologic processor. However, this is not necessarily the case for other RFCs. For RFCs where MAPE data is not required, the MAPE-related configuration is not necessary. The configuration can be done by modifying the current merge modules of the forecast group (KEYINF) where the test locations are contained.

- *cd <test_root>/Config/ModuleConfigFiles/preprocessing/*
- *cp KEYINF_MergeMAP_UpdateStates.xml HEFS_MergeMAP_UpdateStates.xml*
- *cp KEYINF_MergeMAPE_UpdateStates.xml HEFS_MergeMAPE_UpdateStates.xml*
- *cp KEYINF_MergeMAT_UpdateStates.xml HEFS_MergeMAT_UpdateStates.xml*
- in HEFS_MergeMAP(/MAPE/MAT)_UpdateStates.xml files, update all the locationSetId to be "<locationSetId>Catchments_HEFS</locationSetId>" (with quotation mark removed)
- *cd <test_root>/Config/RegionConfigFiles/*
- *nedit ModuleInstanceDescriptors.xml*
- add the following content to the file:

```

<moduleInstanceDescriptor id="HEFS_MergeMAT_UpdateStates">
  <moduleId>TransformationModule</moduleId>
</moduleInstanceDescriptor>
<moduleInstanceDescriptor id="HEFS_MergeMAP_UpdateStates">
  <moduleId>TransformationModule</moduleId>
</moduleInstanceDescriptor>
<moduleInstanceDescriptor id="HEFS_MergeMAPE_UpdateStates">
  <moduleId>TransformationModule</moduleId>
</moduleInstanceDescriptor>

```

2.3.2.3 Update carry over module files related to hydrologic models

This step updates the merge module instance ID in the carryover module files related to the hydrologic models.

- *cd <test_root>/Config/ModuleConfigFiles/XX/*
- *nedit SNOW17_XX_XX_UpdateStates.xml*
- *replace KEYINF_MergeMAT_UpdateStates.xml with HEFS_MergeMAT_UpdateStates.xml;*
replace KEYINF_MergeMAP_UpdateStates.xml with HEFS_MergeMAP_UpdateStates.xml
- *nedit SACSMA_XX_XX_UpdateStates.xml*
- *replace KEYINF_MergeMAPE_UpdateStates.xml with HEFS_MergeMAPE_UpdateStates.xml*

where “XX” stands for the name of the test location (CBNK1/BLKO2 when used in file name; cbnk1/blko2 when used as the name of a directory).

2.3.3 Create/update forecast workflow and relevant configuration files

Configuring the forecast workflow and its associated configuration files includes:

- 1) importing the MEFP hindcast data via the import workflow ImportMEFP.xml;
- 2) pre-processing the imported MEFP hindcasts by the workflow HEFS_PreProcessing_Forecast.xml;
- 3) configuring the hydrologic models to ingest the processed MEFP hindcasts and produce streamflow hindcasts by updating the merge module instance ID in SNOW17(SACSMA)_XX_XX_Forecast.xml; and
- 4) exporting the streamflow hindcasts through the export workflow HEFS_Flow_Forecast.xml.

2.3.3.1 Import MEFP hindcasts

This step involves creating a module instance to copy the MEFP hindcasts from the data directory to the import directory at each issue time (refer to Section 2.2), and creating a workflow to execute an import module instance to import the data from the

import directory. As a side note, for the test locations, MEFP-produced FMAP and FMAT data are with a forecast length of 14 days. Accordingly, the streamflow hindcasts to be produced will also have a forecast length of 14 days. The forecast length will be defined before conducting the hindcast run, as to be described in Section 3.2.

- `cd <test_root>/Config/ModuleConfigFiles/import`
- `edit Import_MEFP_copy.xml`

In the file opened, add the content in the table below:

Standard Location: <test_root>/Config/ModuleConfigFiles/import/

Content of Import_MEFP_copy.xml
<pre> <?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 /awips/chps_local/schemas/generalAdapterRun.xsd"> <general> <description>copy files from backup to import dir</description> <rootDir>%REGION_HOME%/Import</rootDir> <workDir>%ROOT_DIR%</workDir> <exportDir>%ROOT_DIR%</exportDir> <importDir>%ROOT_DIR%</importDir> <dumpFileDir>%REGION_HOME%/DumpFiles</dumpFileDir> <dumpDir>%ROOT_DIR%</dumpDir> <diagnosticFile>%ROOT_DIR%/diag_placeholder.xml</diagnosticFile> <time0Format>yyyyMMdd</time0Format> </general> <activities> <executeActivities> <executeActivity> <command> <executable>%ROOT_DIR%/Copyfiles</executable> </command> <arguments> <argument>%ROOT_DIR%/copy_cardfiles_hindcast_MEFP</argument> <argument>%ROOT_DIR%/cardfiles_hindcast_MEFP</argument> <argument>%TIME0%</argument> </arguments> <timeOut>100000</timeOut> </executeActivity> </executeActivities> </activities> </generalAdapterRun> </pre>

Now create the import workflow:

- `cd <test_root>/Config/WorkflowFiles/hefs`
- `nedit ImportMEFP.xml`

In the file opened, add the content in the table below:

Standard Location: <test_root>/Config/ModuleConfigFiles/import/

Content of ImportMEFP.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>ImportDataCardHindcastMEFP</moduleInstanceId>
  </activity>
</workflow>
```

Next create the import module instance:

- `cd <test_root>/Config/ModuleConfigFiles/imort`
- `nedit ImportDataCardHindcastMEFP.xml`

In the file opened, add the content in the table below:

Standard Location: <test_root>/Config/ModuleConfigFiles/import/

Content of ImportDataCardHindcastMEFP.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<timeSeriesImportRun 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/timeSeriesImportRun.xsd">
  <import>
    <general>
      <importType>NWS-DATACARD-HINDCAST</importType>
      <folder>${IMPORT_FOLDER_DATACARD_HINDCAST}$</folder>
      <idMapId>IdImportDataCardHindcastMEFP</idMapId>
      <unitConversionsId>ImportEnglishUnits</unitConversionsId>
      <missingValue>-999.000</missingValue>
      <importTimeZone>
        <timeZoneOffset>+00:00</timeZoneOffset>
      </importTimeZone>
    </general>
    <timeSeriesSet>
      <moduleInstanceId>ImportDataCardHindcastMEFP</moduleInstanceId>
      <valueType>scalar</valueType>
      <parameterId>FMAP</parameterId>
      <locationSetId>Catchments_HEFS</locationSetId>
      <timeSeriesType>external historical</timeSeriesType>
      <timeStep unit="hour" multiplier="6"/>
      <readWriteMode>add originals</readWriteMode>
      <synchLevel>1</synchLevel>
      <ensembleId>EPP3</ensembleId>
    </timeSeriesSet>
  </import>
  <timeSeriesSet>
    <moduleInstanceId>ImportDataCardHindcastMEFP</moduleInstanceId>
```

```

        <valueType>scalar</valueType>
        <parameterId>FMAT</parameterId>
        <locationSetId>Catchments_HEFS</locationSetId>
        <timeSeriesType>external historical</timeSeriesType>
        <timeStep unit="hour" multiplier="6"/>
        <readWriteMode>add originals</readWriteMode>
        <synchLevel>1</synchLevel>
        <ensembleId>EPP3</ensembleId>
    </timeSeriesSet>
</import>
</timeSeriesImportRun>

```

And create an Id map file for the import module instance:

- *cd <test_root>/Config/IdMapFiles/*
- *nedit IdImportDataCardHindcastMEFP.xml*

In the file opened, add the content in the table below:

Standard Location: <test_root>/Config/IdMapFiles/

Content of IdImporDataCardHindcasttMEFP.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<idMap version="1.1" 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/idMap.xsd">
    <map internalParameter="FMAP" internalLocation="BLKO2" externalParameter="MAP" externalLocation="blko2"/>
    <map internalParameter="FMAP" internalLocation="CBNK1" externalParameter="MAP" externalLocation="cbnk1"/>
    <map internalParameter="FMAT" internalLocation="BLKO2" externalParameter="MAT" externalLocation="blko2"/>
    <map internalParameter="FMAT" internalLocation="CBNK1" externalParameter="MAT" externalLocation="cbnk1"/>
    <enableOneToOneMapping/>
</idMap>

```

Last register the workflow and module instances just created:

- *cd <test_root>/Config/RegionConfigFiles/*
- *nedit WorkflowDescriptors.xml*
- **add the following content**

```

<workflowDescriptor id="ImportMEFP" forecast="false" visible="true" name="Import MEFP" allowApprove="false">
    <description>Import MEFP derived data</description>
</workflowDescriptor>

```

- *nedit ModuleInstanceDescriptors.xml*

add the following lines:

```

<moduleInstanceDescriptor id="Import_MEFP_copy">
    <description>copy files from backup to import dir</description>
    <moduleId>GeneralAdapter</moduleId>
</moduleInstanceDescriptor>
<moduleInstanceDescriptor id="ImportDataCardHindcastMEFP">
    <description>Imports Data cards</description>
    <moduleId>TimeSeriesImportRun</moduleId>
</moduleInstanceDescriptor>

```

Note in the import module instance, “EPP3” is used as the ensemble ID, which is the equivalent of “MEFP”. When configuring locations in other RFCs, “MEFP” can be used as the ensemble ID.

2.3.3.2 Pre-process MEFP hindcasts

This step merges MEFP hindcasts with historical MAP/MAT observations. Similar to Section 2.3.2.2, this can be done by modifying the current merge modules of the forecast group (KEYINF) where the test locations are contained.

- *cd <test_root>/Config/ModuleConfigFiles/preprocessing/*
- *cp KEYINF_MergeMAP_Forecast.xml HEFS_MergeMAP_Forecast.xml*
- *cp KEYINF_MergeMAPE_Forecast.xml HEFS_MergeMAPE_Forecast.xml*
- *cp KEYINF_MergeMAT_Forecast.xml HEFS_MergeMAT_Forecast.xml*
- in HEFS_MergeMAP(MAPE/MAT)_Forecast.xml files, update all the locationSetId to be “<locationSetId>Catchments_HEFS</locationSetId >” (with quotation mark removed)
- *cd <test_root>/Config/RegionConfigFiles/*
- *nedit ModuleInstanceDescriptors.xml*
- add the following content to the file:

```
<moduleInstanceDescriptor id="HEFS_MergeMAT_Forecast">
  <moduleId>TransformationModule</moduleId>
  <simulatedHistoricalModuleInstanceId>HEFS_MergeMAT_UpdateStates</simulatedHistoricalModuleInstanceId>
</moduleInstanceDescriptor>
<moduleInstanceDescriptor id="HEFS_MergeMAP_Forecast">
  <moduleId>TransformationModule</moduleId>
  <simulatedHistoricalModuleInstanceId>HEFS_MergeMAP_UpdateStates</simulatedHistoricalModuleInstanceId>
</moduleInstanceDescriptor>
<moduleInstanceDescriptor id="HEFS_MergeMAPE_Forecast">
  <moduleId>TransformationModule</moduleId>
  <simulatedHistoricalModuleInstanceId>HEFS_MergeMAPE_UpdateStates</simulatedHistoricalModuleInstanceId>
</moduleInstanceDescriptor>
```

Next, create a preprocessing workflow to calls these three merge module instances created:

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

In the file opened, add the content in the table below:

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

Content of HEFS_PreProcessing_Forecast.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>HEFS_MergeMAT_Forecast</moduleInstanceId>
  </activity>
  <ensemble>
    <ensembleId>EPP3</ensembleId>
    <runInLoop>true</runInLoop>
  </ensemble>
  <activity>
    <runIndependent>true</runIndependent>
    <moduleInstanceId>HEFS_MergeMAP_Forecast</moduleInstanceId>
  </activity>
  <ensemble>
    <ensembleId>EPP3</ensembleId>
    <runInLoop>true</runInLoop>
  </ensemble>
  <activity>
    <runIndependent>true</runIndependent>
    <moduleInstanceId>HEFS_MergeMAPE_Forecast</moduleInstanceId>
  </activity>
  <ensemble>
    <ensembleId>EPP3</ensembleId>
    <runInLoop>true</runInLoop>
  </ensemble>
</workflow>
```

Last register the workflow:

- *cd <test_root>/Config/RegionConfigFiles/*
- *edit WorkflowDescriptors.xml*
- **add the following line to the file opened:**

```
<workflowDescriptor id="HEFS_PreProcessing_Forecast" name="HEFS_PreProcessing_Forecast" forecast="true"
visible="false"/>
```


2.3.3.3 Update forecast module files related to hydrologic models

This step updates the merge module instance ID (using the merge module created for the test locations exclusively) in the forecast module files related to the hydrologic models:

- `cd <test_root>/Config/ModuleConfigFiles/XX/`
- `nedit SNOW17_XX_XX_Forecast.xml`
- replace KEYINF_MergeMAT_Forecast with HEFS_MergeMAT_Forecast; replace KEYINF_MergeMAP_Forecast with HEFS_MergeMAP_Forecast
- `nedit SACSMA_XX_XX_Forecast.xml`
- replace KEYINF_MergeMAPE_Forecast with HEFS_MergeMAPE_Forecast

2.3.3.4 Export streamflow hindcasts

This step creates an export workflow and two corresponding export module instances to export the streamflow hindcasts generated by the hydrologic processor.

- `cd <test_root>/Config/WorkflowFiles/hefs`
- `nedit HEFS_Flow_Export.xml`

In the file opened, add the content in the table below:

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

Content of HEFS_Flow_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>Export_ESP_Hindcast_CBNK1</moduleInstanceId>
  </activity>
  <activity>
    <runIndependent>true</runIndependent>
    <moduleInstanceId>Export_ESP_Hindcast_BLK02</moduleInstanceId>
  </activity>
</workflow>
```

- `cd <test_root>/Config/ModuleConfigFiles/hefs`
- `nedit Export_ESP_Hindcast_XX.xml`

Add the following content (using CBNK1 as an example; for BLKO2, update all “CBNK1” to be “BLKO2”) to the file opened:

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

Content of Export_ESP_Hindcast_CBNK1.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<timeSeriesExportRun 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/timeSeriesExportRun.xsd">
  <export>
    <general>
      <exportTypeStandard>pi 1.5</exportTypeStandard>
      <folder>$EXPORT_FOLDERS$/CBNK1</folder>
      <exportFileName>
        <name>_CBNK1_GFS_ESP.</name>
        <prefix>
          <timeZeroFormattingString>yyyyMMddHH</timeZeroFormattingString>
        </prefix>
        <suffix>
          <simpleString>xml</simpleString>
        </suffix>
      </exportFileName>
      <idMapId>IdExportPIXML</idMapId>
      <unitConversionsId>importEnglishUnits</unitConversionsId>
      <exportMissingValue>-999</exportMissingValue>
    </general>
    <timeSeriesSet>
      <moduleInstanceId>ADJUSTQ_CBNK1_CBNK1_Forecast</moduleInstanceId>
      <valueType>scalar</valueType>
      <parameterId>QINE</parameterId>
      <locationId>CBNK1</locationId>
      <timeSeriesType>simulated forecasting</timeSeriesType>
      <timeStep unit="hour" multiplier="6"/>
      <relativeViewPeriod unit="day" start="0" end="14"/>
      <readWriteMode>add originals</readWriteMode>
      <ensembleId>EPP3</ensembleId>
      <ensembleMemberIndexRange start='1951' end='1998'/>
    </timeSeriesSet>
  </export>
</timeSeriesExportRun>
```

Now register the workflow and module instances created:

- *cd* <test_root>/Config/RegionConfigFiles/
- *nedit WorkflowDescriptors.xml*
- add the following lines to the file opened:

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

- *nedit ModuleInstanceDescriptors.xml*
- add the following lines to the file opened:

```

<moduleInstanceDescriptor id="Export_ESP_Hindcast_CBNK1">
  <description>Export ESP Streamflow Hindcasts CBNK1</description>
  <moduleId>TimeSeriesExportRun</moduleId>
</moduleInstanceDescriptor>
<moduleInstanceDescriptor id="Export_ESP_Hindcast_BLK02">
  <description>Export ESP Streamflow Hindcasts BLK02</description>
  <moduleId>TimeSeriesExportRun</moduleId>
</moduleInstanceDescriptor>

```

Note the file name Export_ESP_Hindcast_XX.xml is specific to the ABRFC test locations considered here. For other locations, XX_HEFS_Export.xml can be used for consistency.

2.3.3.5 Create forecast workflow

This step creates the parent workflow HEFS_Flow_Forecast.xml. This workflow

- 1) calls the import workflow ImportMEFP.xml to import the MEFP hindcasts;
- 2) executes the pre-processing workflow HEFS_PreProcessing_Forecast.xml to merge historical observations into MEFP hindcasts imports;
- 3) runs hydrologic models (via workflows XX_Flow_Forecast.xml) to produce streamflow hindcasts; and
- 4) exports the hindcasts produced through the export workflow HEFS_Flow_Export.xml.

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

Add the following content to the file opened:

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

Content of HEFS_Flow_Forecast.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>Import_MEFP_copy</moduleInstanceId>
  </activity>
  <activity>
    <runIndependent>true</runIndependent>
    <workflowId>ImportMEFP</workflowId>
  </activity>
  <activity>
    <runIndependent>false</runIndependent>
    <workflowId>HEFS_PreProcessing_Forecast</workflowId>
  </activity>
  <activity>
    <runIndependent>false</runIndependent>
    <workflowId>CBNK1_Flow_Forecast</workflowId>
    <ensemble>
      <ensembleId>EPP3</ensembleId>
      <ensembleMemberIndexRange start='1951' end='1998'/>
    </ensemble>
  </activity>
</workflow>

```

```

        <runInLoop>true</runInLoop>
    </ensemble>
</activity>
<activity>
    <runIndependent>false</runIndependent>
    <workflowId>BLKO2_Flow_Forecast</workflowId>
    <ensemble>
        <ensembleId>EPP3</ensembleId>
        <ensembleMemberIndexRange start='1951' end='1998'/>
        <runInLoop>true</runInLoop>
    </ensemble>
</activity>
<activity>
    <runIndependent>false</runIndependent>
    <workflowId>HEFS_Flow_Export</workflowId>
</activity>
</workflow>

```

And register the workflow:

- *cd <test_root>/Config/RegionConfigFiles/*
- *nedit WorkflowDescriptors.xml*
- add the following lines to the file:

```

<workflowDescriptor id="HEFS_Flow_Forecast" name="HEFS_Flow_Forecast" forecast="true" visible="true"
allowApprove="true" autoApprove="true">
  <description>HEFS Flow Forecast using updated states</description>
  <cardinalTimeStep id="12Z"/>
  <runExpiryTime unit="day" multiplier="100"/>
</workflowDescriptor>

```

2.4 Configuration of EnsPost

It is assumed that the EnsPost is calibrated in advance. For detailed description on EnsPost and calibration of EnsPost parameters, the readers are referred to EnsPost User's Manual in HEFS 0.1.2 Release (OHD, 2012). The calibrated EnsPost parameters need to be stored at <test_root>/Config/ModuleDataSetFiles/hefs/. Then the configuration of EnsPost includes creating an EnsPost forecast module instance for each location (ENSPOST_XX_Forecast.xml) and an export module instance (Export_EnsPost_XX.xml) to export the post-processed streamflow hindcasts for user-defined location(s). Figure 3 shows the files to be created or modified.

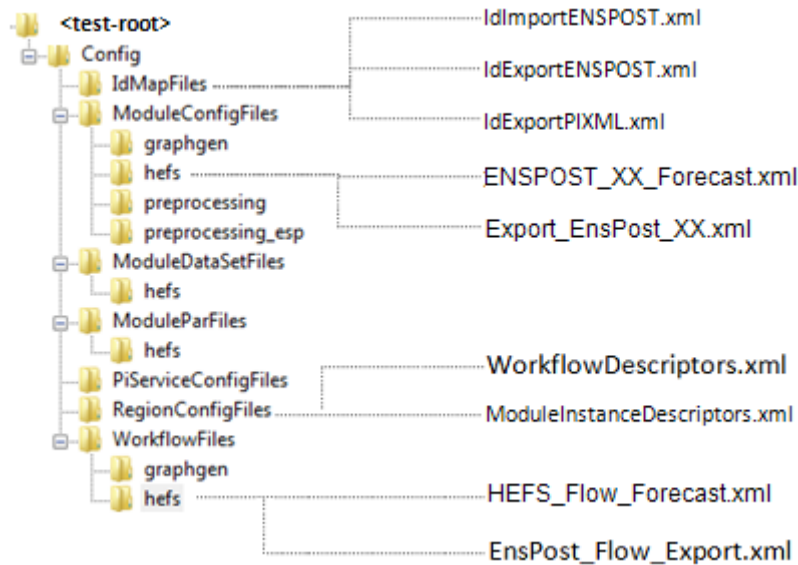


Figure 3. File structure of EnsPost configuration

2.4.1 Create EnsPost forecast module instances

This involves creating a module instance for each test location to prepare inputs to the EnsPost, execute the EnsPost, and produce the post-processed streamflow hindcasts.

- `cd <test_root>/Config/ModuleConfigFiles/hefs/`
- `nedit ENSPOST_CBNK1_Forecast.xml`
- add the following content to the file:

Add the following content to the file opened:

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

Content of ENSPOST_CBNK1_Forecast.xml

```
<generalAdapterRun 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/generalAdapterRun.xsd">
  <general>
    <description>EnsPost CBNK1</description>
    <piVersion>1.5</piVersion>
    <rootDir>${MODELDIR}/hefs/ens_post</rootDir>
    <workDir>%ROOT_DIR%/work</workDir>
    <exportDir>%ROOT_DIR%/input</exportDir>
    <exportDataSetDir>%ROOT_DIR%/work/parameters</exportDataSetDir>
    <exportIdMap>IdExportENSPOST</exportIdMap>
    <importDir>%ROOT_DIR%/output</importDir>
    <importIdMap>IdImportENSPOST</importIdMap>
    <dumpFileDir>${GA_DUMPFILERDIR}</dumpFileDir>
    <dumpDir>%ROOT_DIR%</dumpDir>
    <diagnosticFile>%ROOT_DIR%/output/diag.xml</diagnosticFile>
  </general>
  <activities>
    <startUpActivities>
```

```

<purgeActivity>
  <filter>%ROOT_DIR%/work/*.*</filter>
</purgeActivity>
<purgeActivity>
  <filter>%ROOT_DIR%/input/*.*</filter>
</purgeActivity>
<purgeActivity>
  <filter>%ROOT_DIR%/output/*.*</filter>
</purgeActivity>
</startUpActivities>
<exportActivities>
  <exportTimeSeriesActivity>
    <exportFile>inputs.xml</exportFile>
    <timeSeriesSets>
      <timeSeriesSet>
        <moduleInstancelId>ImportDataCard</moduleInstancelId>
        <valueType>scalar</valueType>
        <parameterId>QME</parameterId>
        <locationId>CBNK1</locationId>
        <timeSeriesType>external historical</timeSeriesType>
        <timeStep unit="hour" multiplier="24"/>
        <relativeViewPeriod unit="hour" start="-240" startOverrutable="true" end="0" endOverrutable="true"/>
        <readWriteMode>add originals</readWriteMode>
      </timeSeriesSet>
      <timeSeriesSet>
        <moduleInstancelId>UNITHG_CBNK1_CBNK1_Forecast</moduleInstancelId>
        <valueType>scalar</valueType>
        <parameterId>SQIN</parameterId>
        <locationId>CBNK1</locationId>
        <timeSeriesType>simulated forecasting</timeSeriesType>
        <timeStep unit="hour" multiplier="6"/>
        <relativeViewPeriod unit="hour" start="6" end="120" endOverrutable="true"/>
        <readWriteMode>read only</readWriteMode>
        <ensembleId>EPP3</ensembleId>
        <ensembleMemberIndexRange start="1951" end="1998"/>
      </timeSeriesSet>
    </timeSeriesSets>
  </exportTimeSeriesActivity>
  <exportDataSetActivity>
    <moduleInstancelId>ENSPOST_CBNK1_Forecast</moduleInstancelId>
  </exportDataSetActivity>
  <exportParameterActivity>
    <fileName>params.xml</fileName>
    <moduleInstancelId>XEFS_Dummy_Parameters</moduleInstancelId>
  </exportParameterActivity>
  <exportRunFileActivity>
    <exportFile>%ROOT_DIR%/run_info.xml</exportFile>
    <properties>
      <string key="model" value="ohd.hseb.hefs.enspost.adapter.HEFSEnsPostModelAdapter"/>
      <int key="printDebugInfo" value="0"/>
      <string key="legacyLocation" value="$XEFSBINDIR$"/>
      <string key="errorModel" value="ER2D"/>
    </properties>
  </exportRunFileActivity>
</exportActivities>
</executeActivities>

```

```

<executeActivity>
  <command>
    <className>ohd.hseb.hefs.utils.adapter.HEFSModelAdapter</className>
    <binDir>${HEFSBINDIR$}</binDir>
  </command>
  <arguments>
    <argument>%ROOT_DIR%/run_info.xml</argument>
  </arguments>
  <timeOut>60000</timeOut>
</executeActivity>
</executeActivities>
<importActivities>
  <importTimeSeriesActivity>
    <importFile>outputs.xml</importFile>
    <timeSeriesSets>
      <timeSeriesSet>
        <moduleInstancelId>ENSPOST_CBK1_Forecast</moduleInstancelId>
        <valueType>scalar</valueType>
        <parameterId>SQIN</parameterId>
        <locationId>CBK1</locationId>
        <timeSeriesType>simulated forecasting</timeSeriesType>
        <timeStep unit="hour" multiplier="6"/>
        <readWriteMode>add originals</readWriteMode>
        <ensembleId>HEFSENSPOST</ensembleId>
        <ensembleMemberIndexRange start="1951" end="1998"/>
      </timeSeriesSet>
    </timeSeriesSets>
  </importTimeSeriesActivity>
</importActivities>
</activities>
</generalAdapterRun>

```

- *nedit ENSPOST_BK02_Forecast.xml*
- add the same content as added to ENSPOST_CBK1_Forecast.xml, but a) changing all “CBK1” to be “BK02”; and b) using the following as the content of <exportTimeSeriesActivity>:

```

<exportTimeSeriesActivity>
  <exportFile>inputs.xml</exportFile>
  <timeSeriesSets>
    <timeSeriesSet>
      <moduleInstancelId>ImportDataCard</moduleInstancelId>
      <valueType>scalar</valueType>
      <parameterId>QME</parameterId>
      <locationId>BK02</locationId>
      <timeSeriesType>external historical</timeSeriesType>
      <timeStep unit="hour" multiplier="24"/>
      <relativeViewPeriod unit="hour" start="-240" startOverrutable="true" end="0" endOverrutable="true"/>
      <readWriteMode>add originals</readWriteMode>
    </timeSeriesSet>
    <timeSeriesSet>
      <moduleInstancelId>ADDSUB_BK02_ROUTED_Forecast</moduleInstancelId>
      <valueType>scalar</valueType>
      <parameterId>SQIN</parameterId>
      <locationId>BK02</locationId>
    </timeSeriesSet>
  </timeSeriesSets>

```

```

    <timeSeriesType>simulated forecasting</timeSeriesType>
    <timeStep unit="hour" multiplier="6"/>
    <relativeViewPeriod unit="hour" start="6" end="120" endOverrutable="true"/>
    <readWriteMode>read only</readWriteMode>
    <ensembleId>EPP3</ensembleId>
    <ensembleMemberIndexRange start="1951" end="1998"/>
  </timeSeriesSet>
</timeSeriesSets>
</exportTimeSeriesActivity>

```

Now register both module instances:

- `cd <test_root>/Config/RegionConfigFiles/`
- `nedit ModuleInstanceDescriptors.xml`
- add the following lines to the file:

```

<moduleInstanceDescriptor id="ENSPOST_CBK1_Forecast">
  <moduleId>GeneralAdapter</moduleId>
</moduleInstanceDescriptor>
<moduleInstanceDescriptor id="ENSPOST_BLK02_Forecast">
  <moduleId>GeneralAdapter</moduleId>
</moduleInstanceDescriptor>

```

Last, create an import IdMap file (applied in above two EnsPost forecast module instances) to enable one to one mapping between the internal parameters and external parameter for the variables to be input to EnsPost:

- `cd <test_root>/Config/IdMapFiles/`
- `nedit IdImportENSPOST.xml`

Add the following content to the file opened:

Standard Location: <test_root>/Config/IdMapFiles/

Content of IdImportENSPOST.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<idMap version="1.1" 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/idMap.xsd">
  <enableOneToOneMapping/>
</idMap>

```

Similarly, an export IdMap files needs to be created to map the internal parameters to external parameters for the output variables of EnsPost. This file has the same content as the import IdMap file. It can be created as:

- `cp IdImportENSPOST.xml IdExportENSPOST.xml`

Note ENSPOST_XX_Forecast.xml is used as the module instance name. For other RFCs, XX_ENSPOST_Forecast.xml is typically used.

2.4.2 Create EnsPost export workflow and module instances

This step involves creating an export module instance for each test location to export the data produced by the EnsPost, and then creating a workflow to execute the module instances.

First, create the export module instances:

- `cd <test_root>/Config/ModuleConfigFiles/hefs/`
- `nedit Export_EnsPost_CBNK1.xml`

Add the following content to the file opened:

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

Content of Export_EnsPost_CBNK1.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<timeSeriesExportRun 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/timeSeriesExportRun.xsd">
  <export>
    <general>
      <exportTypeStandard>pi 1.5</exportTypeStandard>
      <folder>$EXPORT_FOLDER$</folder>
      <exportFileName>
        <name>_CBNK1_EnsPost.</name>
        <prefix>
          <timeZeroFormattingString>yyyyMMddHH</timeZeroFormattingString>
        </prefix>
        <suffix>
          <simpleString>xml</simpleString>
        </suffix>
      </exportFileName>
      <idMapId>IdExportPIXML</idMapId>
      <unitConversionsId>importEnglishUnits</unitConversionsId>
      <exportMissingValue>-999</exportMissingValue>
    </general>
    <timeSeriesSet>
      <moduleInstanceId>ENSPOST_CBNK1_Forecast</moduleInstanceId>
      <valueType>scalar</valueType>
      <parameterId>SQIN</parameterId>
      <locationId>CBNK1</locationId>
      <timeSeriesType>simulated forecasting</timeSeriesType>
      <timeStep unit="hour" multiplier="6"/>
        <relativeViewPeriod unit="day" start="0" end="14"/>
      <readWriteMode>add originals</readWriteMode>
      <ensembleId>HEFSENSPOST</ensembleId>
      <ensembleMemberIndexRange start="1951" end="1998"/>
    </timeSeriesSet>
  </export>
</timeSeriesExportRun>
```

- *cp Export_ EnsPost_CBNK1.xml Export_ EnsPost_BLK02.xml*
- *nedit Export_ EnsPost_BLK02.xml*
- change all CBNK1 to be BLK02

Next, register these two module instances:

- *cd <test_root>/Config/RegionConfigFiles/*
- *nedit ModuleInstanceDescriptors.xml*
- add the following lines to the file:

```
<moduleInstanceDescriptor id="Export_EnsPost_CBNK1">
  <description>Export EnsPost-processed Hindcasts CBNK1</description>
  <moduleId>TimeSeriesExportRun</moduleId>
</moduleInstanceDescriptor>
<moduleInstanceDescriptor id="Export_EnsPost_BLK02">
  <description>Export EnsPost-processed Hindcasts BLK02</description>
  <moduleId>TimeSeriesExportRun</moduleId>
</moduleInstanceDescriptor>
```

Now create the IdExport file applied in two export module instances (if not existing). The file has the same content as the IdImportENSPOST.xml created in Section 2.4.1.

- *cd <test_run>/Config/IdMapFiles/*
- *cp IdImportENSPOST.xml IdExportPIXML.xml*

Then create the export workflow:

- *cd <test_run>/Config/WorkflowFiles/hefs/*
- *nedit EnsPost_Flow_Export.xml*

Add the following content to the file opened:

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

Content of EnsPost_Flow_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>Export_EnsPost_CBNK1</moduleInstanceId>
  </activity>
  <activity>
    <runIndependent>true</runIndependent>
    <moduleInstanceId>Export_EnsPost_BLK02</moduleInstanceId>
  </activity>
</workflow>
```

Note that if the user is interested in only exporting the data for selected locations, then the user will only need to include the export module instances for those locations as the activities in this export workflow.

Last, register the workflow:

- `cd <test_run>/Config/RegionConfigFiles/`
- `nedit WorkflowDescriptors.xml`
- add the following lines to the file:

```
<workflowDescriptor id="EnsPost_Flow_Export" name="EnsPost Export" forecast="false" visible="true" allowApprove="false"
autoApprove="false">
  <description>EnsPost Export</description>
</workflowDescriptor>
```

Note `Export_EnsPost_XX.xml` is used as the export module instance name here, while `XX_ENSPOST_Export.xml` is typically used in other RFCs.

2.4.3 Add EnsPost forecast and export files to the parent workflow

This involves adding `ENSPOST_XX_Forecast.xml` and `EnsPost_Flow_Export.xml` to the parent workflow `HEFS_Flow_Forecast.xml`:

- `cd <test_run>/Config/WorkflowFiles/hefs/`
- `nedit HEFS_Flow_Forecast.xml`
- add the following content to the file:

```
<activity>
  <runIndependent>>false</runIndependent>
  <moduleInstanceId>ENSPOST_CBNK1_Forecast</moduleInstanceId>
</activity>
<activity>
  <runIndependent>>false</runIndependent>
  <moduleInstanceId>ENSPOST_BLK02_Forecast</moduleInstanceId>
</activity>
<activity>
  <runIndependent>>false</runIndependent>
  <workflowId>EnsPost_Flow_Export</workflowId>
</activity>
```

3. Example Run

As mentioned in Section 2.3, the MEFP-based HEFS hindcasting run includes two major steps. The first step is to produce the initial states of the hydrologic models for the hindcasting runs. The second step is to import MEFP-produced data and generate the streamflow hindcasts by running a single workflow. In the meantime, the EnsPost is run to post-process and exports the generated streamflow hindcasts. Those steps are shown below.

3.1 Produce Initial Model States

The initial states of the hydrologic models are produced by running the hydrologic models with historical observed forcing. These data, which include MAP and MAT, need to be imported via the import workflow (e.g., 'Import Scalars '). If MAPE is specified as input for the rainfall-runoff model (i.e., SAC-SMA), it has to be imported as well. ABRFC uses climatological MAPE, which is generated by the workflow 'Preprocess (Climatology)'.

After importing the observed data into the CHPS database, the workflow (HEFS_Flow_UpdateStates.xml) created in Section 2.3.2.1 can be run to produce initial states. Let's assume we want to generate initial states over a 20-year period (01-01-1979 12:00:00 ~ 01-01-1999 12:00:00) at a constant interval of one day, the following procedures can be applied.

3.1.1 Model spin-up-creating initial states for the start of the hindcast run

The hydrologic models should be warmed up by making a single forecast run of the workflow until the start time of the hindcast period (e.g., let $T_0=01-01-1979\ 12:00:00$). Note an appropriate warm-up period (e.g., longer than one year but generally less than two years) should be defined to allow the models to arrive at an appropriate state at T_0 . For more detailed information on warm-up period, the readers are referred to the document *A Strategy for Science Evaluation of the Hydrologic Ensemble Forecast Service (HEFSv1)* (OHD, 2012). This can be set up in the FEWS manual forecast display as shown in Figure 4. In this example, a 2-year warm-up period is used. Note the "Approve" box should be checked so that the model states from this spin-up run will be saved in the database for later use. There is no need to change the forecast length in the manual forecast display.

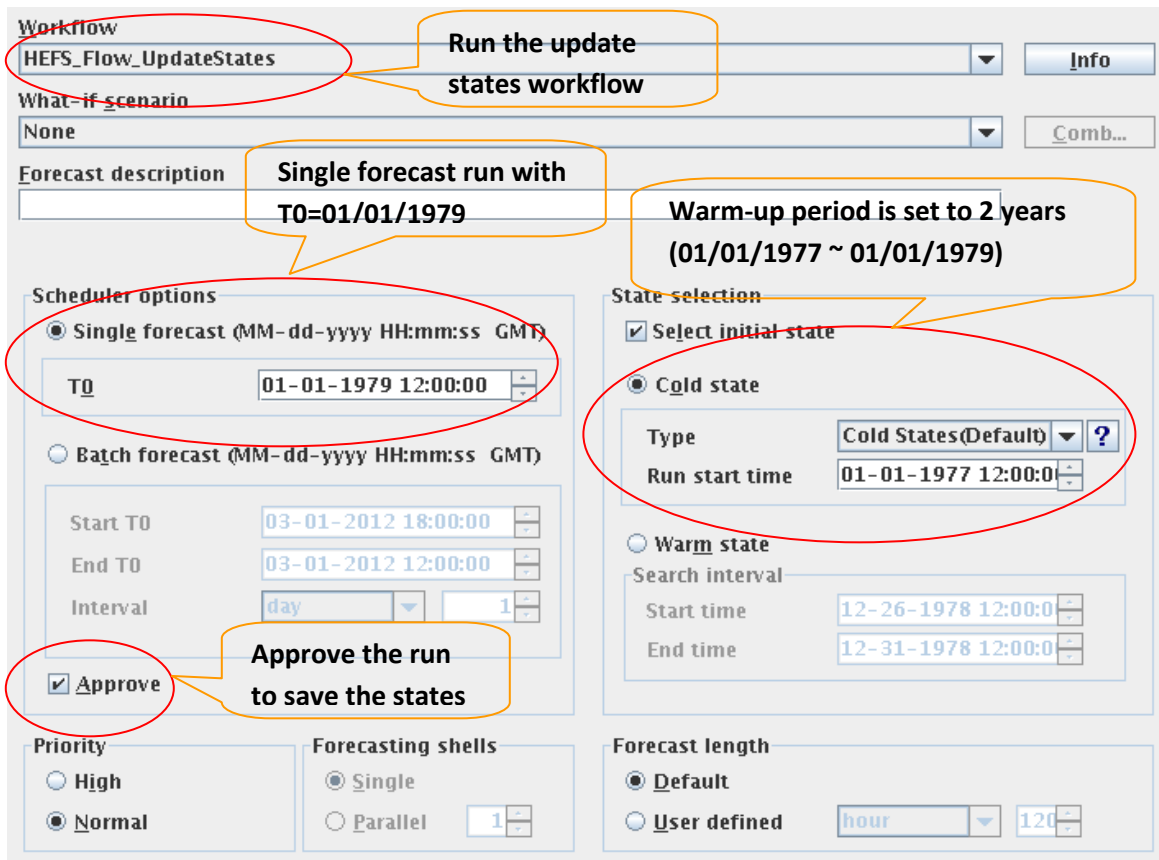


Figure 4. Model spin-up using a single run with a two-year warm-up period

3.1.2 Generate initial states for each hindcast time

To generate initial states for use in a subsequent hindcast run, we can run the update-states workflow (same one as employed in the above subsection) in a batch mode over the hindcast period at a desired interval (e.g., one day), as shown in Figure 5. This will submit a sequence of hindcast runs to the FEWS task manager and these runs will be processed sequentially to generate initial model states over the hindcast period at the defined interval. If it is set up correctly, the first run in the sequence (T0=01-01-1979 12:00:00) will start from the states generated by the spin-up run, the second run (T0=01-02-1979 12:00:00) will use the states generated by the first run, the third run (T0=01-03-1979 12:00:00) will use the states generated by the second run, and so on.

Figure 5. Batch carryover run to generate initial model states

3.2 Generate Streamflow Hindcasts and EnsPost-processed Data

Running the parent workflow (HEFS_Flow_Forecast.xml) in batch mode will produce streamflow hindcasts and call EnsPost to post-process these hindcasts. The detailed instructions are as follows: in Manual Forecast display (refer to Figure 6):

- Select the workflow “HEFS_Flow_Forecast”
- Check “Batch forecast” in Scheduler options; specify Start T0 to be “01-01-1979 12:00:00”, End T0 to be “01-31-1979 12:00:00”, with “1 day” as the interval.
- Define “14 days” as the forecast length
- Press “Run”
- The xml format data (Streamflow hindcasts and EnsPost-processed data) files defined in Export_ESP_Hindcast_CBNK1.xml, Export_ESP_Hindcast_BLKO2.xml, Export_EnsPost_CBNK1.xml and Export_EnsPost_BLKO2.xml will be exported to directory <test_root>/Export/

Workflow
HEFS_Flow_Forecast Info

What-if scenario
 None Comb...

Forecast description

Scheduler options

Single forecast (MM-dd-yyyy HH:mm:ss GMT)

T0

Batch forecast (MM-dd-yyyy HH:mm:ss GMT)

Start T0

End T0

Interval

Approve

Priority
 High
 Normal

State selection (times related to the first T0)

Select initial state

Cold state

Type ?

Run start time

Warm state

Search interval

Start time

End time

Forecast length

Default

User defined

Run the forecast workflow

Batch forecast run over the hindcast period with 1-day interval

Specify the forecast length to be 2 weeks

Figure 6. Run the forecast workflow in a batch mode to generate the hindcasts

The streamflow hindcasts and EnsPost-processed hindcasts can also be visualized via CHPS Database Viewer following the instructions below: inside the Logs Panel of CHPS display:

- Press “F12”, and then select “J”. This will open CHPS Database Viewer, as shown in Figure 7a.

	T0	Dispatch time	Workflow	W...
▲	01-22-1979 12:00:00	03-14-2012 14:59:33	HEFS_Flow_Forecast			heke ▲
▲	01-25-1979 12:00:00	03-14-2012 14:59:33	HEFS_Flow_Forecast			heke
▲	01-18-1979 12:00:00	03-14-2012 14:59:33	HEFS_Flow_Forecast			heke
▲	01-19-1979 12:00:00	03-14-2012 14:59:33	HEFS_Flow_Forecast			heke
▲	01-24-1979 12:00:00	03-14-2012 14:59:33	HEFS_Flow_Forecast			heke
▲	01-01-1979 12:00:00	03-14-2012 14:59:33	HEFS_Flow_Forecast			heke
▲	01-11-1979 12:00:00	03-14-2012 14:59:33	HEFS_Flow_Forecast			heke
▲	01-17-1979 12:00:00	03-14-2012 14:59:33	HEFS_Flow_Forecast			heke
▲	01-23-1979 12:00:00	03-14-2012 14:59:33	HEFS_Flow_Forecast			heke
▲	01-20-1979 12:00:00	03-14-2012 14:59:33	HEFS_Flow_Forecast			heke
▲	01-16-1979 12:00:00	03-14-2012 14:59:33	HEFS_Flow_Forecast			heke
▲	01-10-1979 12:00:00	03-14-2012 14:59:33	HEFS_Flow_Forecast			heke
▲	01-14-1979 12:00:00	03-14-2012 14:59:33	HEFS_Flow_Forecast			heke
▲	01-02-1979 12:00:00	03-14-2012 14:59:33	HEFS_Flow_Forecast			heke
▲	01-31-1979 12:00:00	03-14-2012 14:59:33	HEFS_Flow_Forecast			heke
▲	01-15-1979 12:00:00	03-14-2012 14:59:33	HEFS_Flow_Forecast			heke
▲	01-08-1979 12:00:00	03-14-2012 14:59:33	HEFS_Flow_Forecast			heke

Figure 7a: Database Viewer window

- a) Select a T0 interested (e.g., 01-22-1979 12:00:00; refer to Figure 7b); all time series corresponding to this T0 will show up, as shown in Figure 7b.

	T0	Dispatch time	Workflow	W...
▲	01-22-1979 12:00:00	03-14-2012 14:59:33	HEFS_Flow_Forecast			heke ▲
▲	01-25-1979 12:00:00	03-14-2012 14:59:33	HEFS_Flow_Forecast			heke
▲	01-18-1979 12:00:00	03-14-2012 14:59:33	HEFS_Flow_Forecast			heke
▲	01-19-1979 12:00:00	03-14-2012 14:59:33	HEFS_Flow_Forecast			heke
▲	01-24-1979 12:00:00	03-14-2012 14:59:33	HEFS_Flow_Forecast			heke
▲	01-01-1979 12:00:00	03-14-2012 14:59:33	HEFS_Flow_Forecast			heke
▲	01-11-1979 12:00:00	03-14-2012 14:59:33	HEFS_Flow_Forecast			heke
▲	01-17-1979 12:00:00	03-14-2012 14:59:33	HEFS_Flow_Forecast			heke
▲	01-23-1979 12:00:00	03-14-2012 14:59:33	HEFS_Flow_Forecast			heke
▲	01-20-1979 12:00:00	03-14-2012 14:59:33	HEFS_Flow_Forecast			heke
▲	01-16-1979 12:00:00	03-14-2012 14:59:33	HEFS_Flow_Forecast			heke
▲	01-10-1979 12:00:00	03-14-2012 14:59:33	HEFS_Flow_Forecast			heke
▲	01-14-1979 12:00:00	03-14-2012 14:59:33	HEFS_Flow_Forecast			heke
▲	01-02-1979 12:00:00	03-14-2012 14:59:33	HEFS_Flow_Forecast			heke
▲	01-31-1979 12:00:00	03-14-2012 14:59:33	HEFS_Flow_Forecast			heke
▲	01-15-1979 12:00:00	03-14-2012 14:59:33	HEFS_Flow_Forecast			heke
▲	01-08-1979 12:00:00	03-14-2012 14:59:33	HEFS_Flow_Forecast			heke

moduleInstance	group	para	locatio	locatio...	x	y	timeSe	ense	ensem...	valueType	timeStep
17	9	31	3	3			2	2	48	1	2
LAGK_BLK02_CBNK1_Forecast	Disch...	QINE	CBNK1	CBNK1	-97...	3...	simul...	EPP3	1998	scalar	6 hour
ADJUSTQ_BLK02_BLK02_Forecast	Disch...	QINE	BLK02	BLK02	-97...	3...	simul...	EPP3	1951	scalar	6 hour
ADJUSTQ_BLK02_BLK02_Forecast	Disch...	QINE	BLK02	BLK02	-97...	3...	simul...	EPP3	1952	scalar	6 hour
ADJUSTQ_BLK02_BLK02_Forecast	Disch...	QINE	BLK02	BLK02	-97...	3...	simul...	EPP3	1953	scalar	6 hour
ADJUSTQ_BLK02_BLK02_Forecast	Disch...	QINE	BLK02	BLK02	-97...	3...	simul...	EPP3	1954	scalar	6 hour
ADJUSTQ_BLK02_BLK02_Forecast	Disch...	QINE	BLK02	BLK02	-97...	3...	simul...	EPP3	1955	scalar	6 hour
ADJUSTQ_BLK02_BLK02_Forecast	Disch...	QINE	BLK02	BLK02	-97...	3...	simul...	EPP3	1956	scalar	6 hour
ADJUSTQ_BLK02_BLK02_Forecast	Disch...	QINE	BLK02	BLK02	-97...	3...	simul...	EPP3	1957	scalar	6 hour
ADJUSTQ_BLK02_BLK02_Forecast	Disch...	QINE	BLK02	BLK02	-97...	3...	simul...	EPP3	1958	scalar	6 hour
ADJUSTQ_BLK02_BLK02_Forecast	Disch...	QINE	BLK02	BLK02	-97...	3...	simul...	EPP3	1959	scalar	6 hour
ADJUSTQ_BLK02_BLK02_Forecast	Disch...	QINE	BLK02	BLK02	-97...	3...	simul...	EPP3	1960	scalar	6 hour
ADJUSTQ_BLK02_BLK02_Forecast	Disch...	QINE	BLK02	BLK02	-97...	3...	simul...	EPP3	1961	scalar	6 hour

Figure 7b: Time series window in Database Viewer

- b) Select a module instance of interest (e.g., ADJUSTQ_BLK02_BLK02_Forecast, which produce streamflow hindcasts for BLKO2; refer to Figure 4c); select the time series corresponding to this module from ensemble ID 1951 to ensemble ID 1998, as shown in Figure 7c.

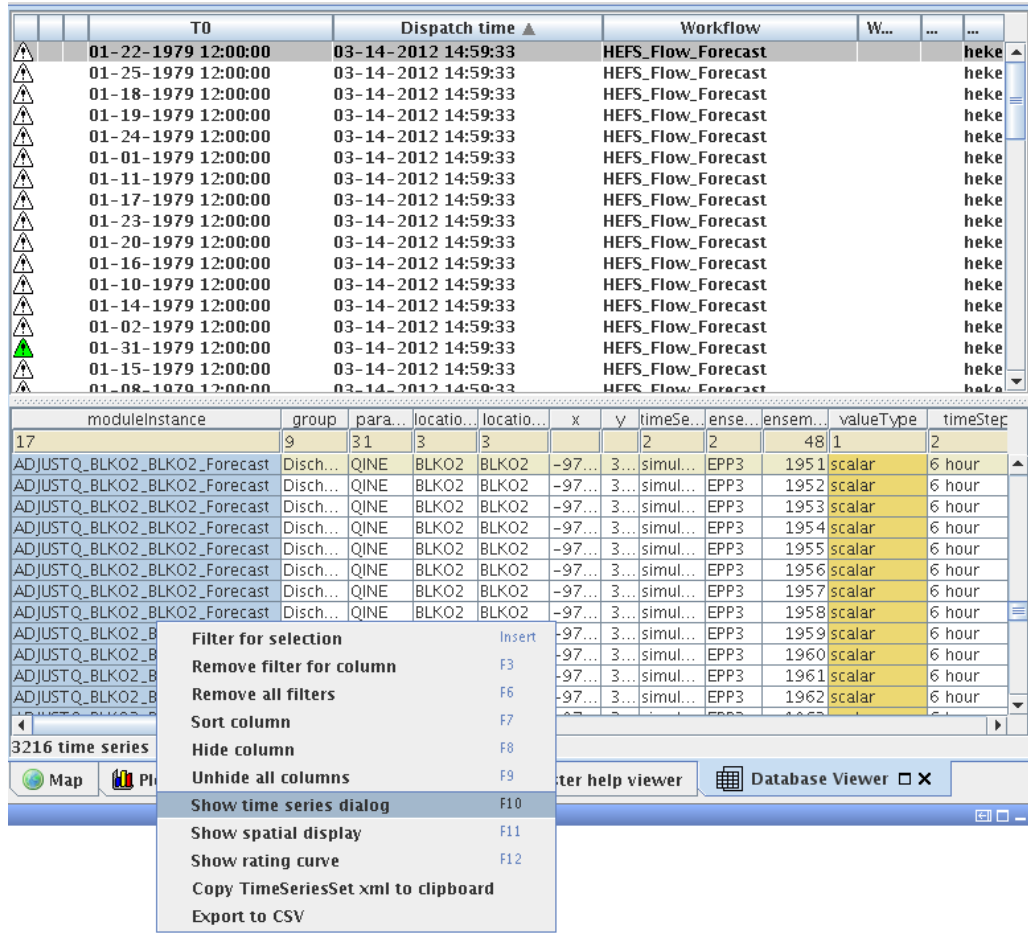


Figure 7c: Select time series to plot

- c) Once the time series of interest are selected, right click and select Show time series dialog F10 (refer to Figure 7c). The selected time series will be plotted, as illustrated in Figure 7d.
- d) Repeat steps b) and c) for other module instances of interest (e.g., ADJUSTQ_CBNK1_CBNK1_Forecast for streamflow hindcasts produced for CBNK1; ENSPOST_BLK02(CBNK1)_Forecast, for EnsPost-processed hindcasts for BLKO2 and CBNK1, respectively).
- e) Repeat steps a) through d) for other T0s of interest.

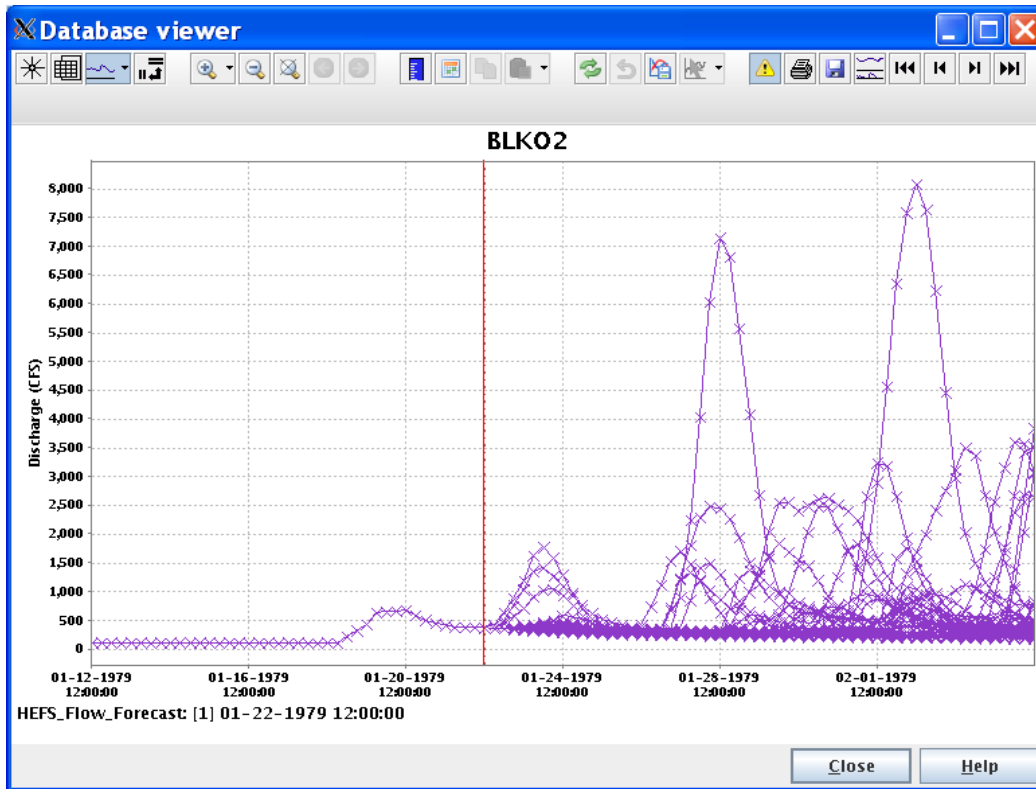


Figure 7d: Plot of streamflow hindcasts at BLKO2

4. Appendix: Generation of MEFP Hindcasts Outside of CHPS

In Section 8 of the Ensemble Preprocessor (EPP3) User's Manual (OHD, 2010), general strategies for running the MEFP/EPP3 in stand-alone mode to generate hindcast ensembles are described. This appendix gives three example scripts illustrating the strategies of generating hindcast ensembles. Before running the scripts, the user needs to first set up the apps-defaults, then place the MEFP/EPP3 executables, parameter files, and other data files in the directories specified by the apps-defaults.

When ensemble generation is done in the hindcast mode, three sets of files are generated. One set is the precipitation and/or temperature hindcast files in the OHD DATACARD format that can be read by ESP. This set of hindcasts can be verified using the OHD Ensemble Verification System (EVS). The second set includes an unformatted file that contains all of the ensemble members as well as the corresponding observations and raw forecasts. This unformatted file is used as input by the MEFP/EPP3 diagnostic evaluation programs `epp3_precip_verify` and/or `epp3_temp_verify`. The third one is a set of ASCII files that can be used for diagnostic

purposes to aid in the development and testing of the MEFP/EPP3 ensemble generation programs.

Example 1

This script is used to generate temperature ensemble hindcasts for two locations in CNRFC for the months of December, January, and February, with hindcast years of 1979 – 2005, using only GFS reforecasts as the input forecast source. The script iterates through these three months, each with a central date specified. On each side of the central date, 15 days are specified so that a hindcast window of 31 days will be formed for each month. Hindcasting will be done for each day in this window as control parameter "offset_days" is set to 1. In the output hindcast files, ensemble traces are associated with the years of available historical observations (1961 - 2005 in this example).

```
=====
fcst_dates="99960116 \
           99960216 \
           99961216"

for fd in $fcst_dates
do

    echo "execution_mode = hindcast" > t_gfs.in
    echo "nrfcdays = 0" >> t_gfs.in
    echo "ngfsdays = 14" >> t_gfs.in
    echo "ncfsdays = 0" >> t_gfs.in
    echo "nclimdays = 0" >> t_gfs.in
    echo "lyr_ensmem = 1961" >> t_gfs.in
    echo "lyr_ensmem = 2005" >> t_gfs.in
    echo "lyr_hindcast = 1979" >> t_gfs.in
    echo "lyr_hindcast = 2005" >> t_gfs.in
    echo "forecast_date = ${fd}" >> t_gfs.in
    echo "offset_number = 15" >> t_gfs.in
    echo "offset_days = 1" >> t_gfs.in
    echo "id = mfac1llf" >> t_gfs.in
    echo "id = mfac1luf" >> t_gfs.in
    echo "end" >> t_gfs.in

    mv t_gfs.in /fs/ensembles/users/lwu/epp3/cnrfc/epp/gfs/control

    /fs/ensembles/users/lwu/epp3/ens_short/bin/epp3_temp_epp t_gfs.in

done
=====
```

Example 2

This script is used to generate precipitation ensemble hindcasts for two locations in CNRFC for the entire year with hindcast years of 1979 - 2005 using RFC QPFs and GFS reforecasts as input forecast sources. The script iterates from January through December with a central date specified for each month. On each side of the central date, 15 days are specified so that a hindcast window of 31 days will be formed for each month. Hindcasting will be done for each day in the window as control parameter "offset_days" is set to 1. Resampled climatology ensembles will be generated for forecast days 15 - 330. In the output hindcast files, ensemble traces are associated with the years of available historical observations (1949 - 2005 in this example).

```
=====
fcst_dates="90090116 \
            90090216 \
            90090316 \
            90090416 \
            90090516 \
            90090616 \
            90090716 \
            90090816 \
            90090916 \
            90091016 \
            90091116 \
            90091216"

for fd in $fcst_dates
do

    echo "execution_mode = hindcast" > p_rfc_gfs_clm.in
    echo "nrfcdays = 2" >> p_rfc_gfs_clm.in
    echo "ngfsdays = 14" >> p_rfc_gfs_clm.in
    echo "ncfsdays = 0" >> p_rfc_gfs_clm.in
    echo "nclimdays = 330" >> p_rfc_gfs_clm.in
    echo "climate_option = resampled" >> p_rfc_gfs_clm.in
    echo "iyр_ensmem = 1949" >> p_rfc_gfs_clm.in
    echo "lyr_ensmem = 2005" >> p_rfc_gfs_clm.in
    echo "iyр_hindcast = 1979" >> p_rfc_gfs_clm.in
    echo "lyr_hindcast = 2005" >> p_rfc_gfs_clm.in
    echo "forecast_date = ${fd}" >> p_rfc_gfs_clm.in
    echo "offset_number = 15" >> p_rfc_gfs_clm.in
    echo "offset_days = 1" >> p_rfc_gfs_clm.in
    echo "id = nfdc1huf" >> p_rfc_gfs_clm.in
    echo "id = nfdc1huf" >> p_rfc_gfs_clm.in
    echo "end" >> p_rfc_gfs_clm.in

    mv p_rfc_gfs_clm.in /fs/ensembles/users/lwu/epp3/cnrfc/epp/gfs/control

    /fs/ensembles/users/lwu/epp3/ens_short/bin/epp3_precip_epp p_rfc_gfs_clm.in

done
=====
```

Example 3

This script is used to generate precipitation ensemble hindcasts for a location in ABRFC for the entire year with hindcast years of 1981 - 2006 using CFSv1 reforecasts as the input forecast source. The script iterates from January through December with a central date specified for each month. On each side of the central date, 15 days are specified so that a hindcast window of 31 days will be formed for each month. Although control parameter "offset_days" is set to 1, hindcasting will only be done for certain days in the specified hindcast window since the CFSv1 reforecasts are available only for 3 blocks of 5 consecutive days ending on the 3rd, 13th, and 23th each month. Because of the 1-day delay between the forecast creation time and the time the forecasts are available in real-time operation, the CFSv1 ensemble hindcasts are made on the 4th, 14th, and 24th each month, to be consistent with the calibration process (ensemble hindcasts are not generated for the other reforecast days in the blocks). In the output hindcast files, ensemble traces are associated with the years of available historical observations (1951 - 1998 in this example).

```
=====
```

```
fcst_dates="90090116 \  
            90090216 \  
            90090316 \  
            90090416 \  
            90090516 \  
            90090616 \  
            90090716 \  
            90090816 \  
            90090916 \  
            90091016 \  
            90091116 \  
            90091216"  
  
for fd in $fcst_dates  
do  
  
    echo "execution_mode = hindcast" > p_cfsv1.in  
    echo "nrfcdays = 0" >> p_cfsv1.in  
    echo "ngfsdays = 0" >> p_cfsv1.in  
    echo "ncfsdays = 240" >> p_cfsv1.in  
    echo "nclimdays = 0" >> p_cfsv1.in  
    echo "cfs_version = cfsv1" >> p_cfsv1.in  
    echo "lyr_ensmem = 1951" >> p_cfsv1.in  
    echo "lyr_ensmem = 1998" >> p_cfsv1.in  
    echo "lyr_hindcast = 1981" >> p_cfsv1.in  
    echo "lyr_hindcast = 2006" >> p_cfsv1.in  
    echo "forecast_date = ${fd}" >> p_cfsv1.in  
    echo "offset_number = 15" >> p_cfsv1.in  
    echo "offset_days = 1" >> p_cfsv1.in  
    echo "ccv_max = 1.5" >> p_cfsv1.in  
    echo "id = blk02" >> p_cfsv1.in
```

```
echo "end" >> p_cfsv1.in  
  
mv p_cfsv1.in /fs/ensembles/users/lwu/epp3/abr/c/epp/gfs/control  
  
/fs/hseb/pda/users/lwu/java/HEFS/trunk/prototypes/epp3/bin/epp3_precip_epp p_cfsv1.in
```

5. References

Ensemble Postprocessor (EnsPost) HEFS Release 0.1.2 User's Manual, HEFS 0.1.2 Part-A Release, OHD, June 2012

Ensemble Pre-processor III (EPP3) Quick Start Guide, OHD, 2010 (Available at <http://www.nws.noaa.gov/oh/XEFS/download.html>)

Meteorological Ensemble Forecast Processor (MEFP) HEFS Release 0.1.2 User's Manual, HEFS 0.1.2 Part-A Release, OHD, June 2012

Overview of the Hydrologic Ensemble Forecast Service (HEFS), HEFS 0.1.2 Part-A Release, OHD, June 2012

A Strategy for Science Evaluation of the Hydrologic Ensemble Forecast Service (HEFSv1), James Brown (james.brown@hydrosolved.com), OHD, May 2012