Deltares

Enabling Delta Life

How to add HECRAS models to CHPS

Configuration Manual

Prepared for:

National Weather Service

How to add HECRAS models to CHPS

Configuration Manual

J.M. Lemans

Report

January 2010

Deltares

Title
How to add HEC-RAS model to CHPS

Client	Project	Pages
National Weather Service	1002089	48

Keywords
HEC-RAS, manual, How to

Summary
This manual describes how an existing HEC-RAS model can be added to a CHPS configuration

Version	Date	Author	Initials	Review	Initials	Approval	Initials
1.0	11-20-2009	Matthijs Lemans					
2.0	01-13-2010	Matthijs Lemans					
3.0	09-11-2010	Matthijs Lemans					
3.1	9-13-2010	Seann Reed					
3.2	02-28-2011	Seann Reed					
3.3	05-31-2011	Seann Reed					
3.4	08-17-2011	Matthijs Lemans					
3.5	09-20-2011	Seann Reed					

Contents

1 Introduction 1
1.1 Versioning for HEC-RAS Adapter 1

2 HEC-RAS environment 2
2.1 Interface between FEWS and HEC-RAS 2
2.2 Local dependent Linux libraries 2
2.3 Environment variable JAVA_HOME 2
2.4 Directory structure 3

3 From HEC-RAS GUI to CHPS 5
3.1 Run HEC-RAS model in the HEC-RAS GUI 6
3.1.1 Open HEC-RAS 6
3.1.2 Initial conditions. 6
3.1.3 Output locations 7
3.1.4 Write restart file 8
3.1.5 Run simulation 9
3.2 Copy the HEC-RAS data to the CHPS environment 11
3.3 General Adapter file in the /Config/ModuleConfigFiles 12
3.4 HECRAS State Management in CHPS 15
3.4.1 Coldstate directory for the HEC-RAS model 15
3.4.2 State Management 15
3.5 IdMapFiles with new HEC-RAS locations 16
3.6 Parameter File in the /Config/ModuleParFiles 18
3.7 UnitConversionsFiles; add conversions for HEC-RAS 18
3.8 HEC-RAS preprocess file and model instability 20
3.8.1 Minimum flow 20
3.8.2 Burn-in profile 20
3.9 HEC-RAS workflows 21
3.10 Register new files and location in the RegionConfigFiles 21
3.11 Add a new directory in /Models 21
3.12 Topology files for Interactive Forecast Display 22
3.13 ModifierTypes 23
3.14 Filters 24
3.15 Longitudinal Profiles 24
3.15.1 Additions to General Adapter 25
3.15.2 Branches.xml file 26
3.15.3 Additions to IdImportHECRAS.xml 28
3.15.4 Addtitions to Filters.xml or DisplayGroups.xml 28
3.16 Add global properties for hecras model and binaries 29
3.17 HEC-RAS Model Performance when run from CHPS on Linux 29

Appendix 1: Description of HEC-RAS data files 30

Appendix 2: General adapter file 31

Appendix 3: Parameter File 34

Appendix 4: Preprocessing **39**

Appendix 5: Filters **42**

1 Introduction

This manual describes how an existing HEC-RAS model can be added to a CHPS configuration. The assumption is that the HEC-RAS model runs in the HEC-RAS GUI without any errors. From there, the several steps to include the model in CHPS are explained. Since the HEC-RAS GUI is Windows based, it is required to have a Windows machine available.
The manual starts with a overview of the HEC-RAS environment.

More information can be found on the Deltares wiki:
http://public.deltares.nl/display/FEWSDOC/HEC-RAS+Model+Adapter

1.1 Versioning for HEC-RAS Adapter

The first official release of the CHPS HEC-RAS Adapter was delivered to OHD on July 29, 2010. This was version 1.0. A fix to the treatment of “IB Stage/Flow” (a.k.a. “Observed Stage and Flow Hydrograph”) was included in a new version delivered on 2/1/2011. This version is 1.01.

Version 1.02 of the adapter will be released in June 2011. This document describes fixes included in version 1.02. These fixes (1) allow for an internal boundary to properly switch from observed stage data to a rating curve when specified, (2) correct the treatment of case for locationIDs in output.xml to be consistent with the HECRAS Unsteady Flow File, and (3) correctly the implementation of inflow multipliers specified in the Unsteady Flow File.

2 HEC-RAS environment

The HEC-RAS model provides the compute engine for running a hydraulic model schematization for a section of a river or a part of a river system. Two adapters, the FEWS General Adapter and the HEC-RAS Model Adapter form the interface between the FEWS Forecasting Shell and the HEC-RAS model.
The HEC-RAS compute engine is, as its name suggests, the component that actually performs the HEC-RAS simulation. This simulation is controlled from the adapters, and all run time data such as initial and boundary conditions, and parameter settings are passed through the adapters from and to the FEWS Forecasting Shell.

2.1 Interface between FEWS and HEC-RAS

The Adapters for HEC-RAS form the interface between the FEWS Forecasting Shell and the HEC-RAS model. The FEWS General Adapter of the Forecasting Shell provides the required run-time data to run HEC-RAS, and calls the HEC-RAS Module Adapter. The data is provided in a standardized XML interface format, the FEWS Published Interface. The HEC-RAS Model Adapter transfers the XML-data into the native HEC-RAS file formats.
Once a HEC-RAS run has been completed, relevant results are passed back by the HEC-RAS Module Adapter to the Forecasting Shell (FEWS General Adapter) in the form of the standardized XML interface format.
A schematic representation of the communication between the Forecasting Shell and the HEC-RAS model via the FEWS Adapter is shown in the Figure 1.

2.2 Local dependent Linux libraries

The following libraries are required when running on Linux:

- Linux-gate.so.1
- Libmtsk.so.1
- Libc.so.6
- Libpthread.so.0
- Libm.so.6
- Librt.so.1
- Libdl.so.2

2.3 Environment variable JAVA_HOME

The HEC-RAS adapter uses the environment variable $JAVA_HOME. For Operator Client and Stand Alone this is set when starting FEWS. For the Forecasting Shell Server please ensure this is set in the environment and that refers to the correct Java version.

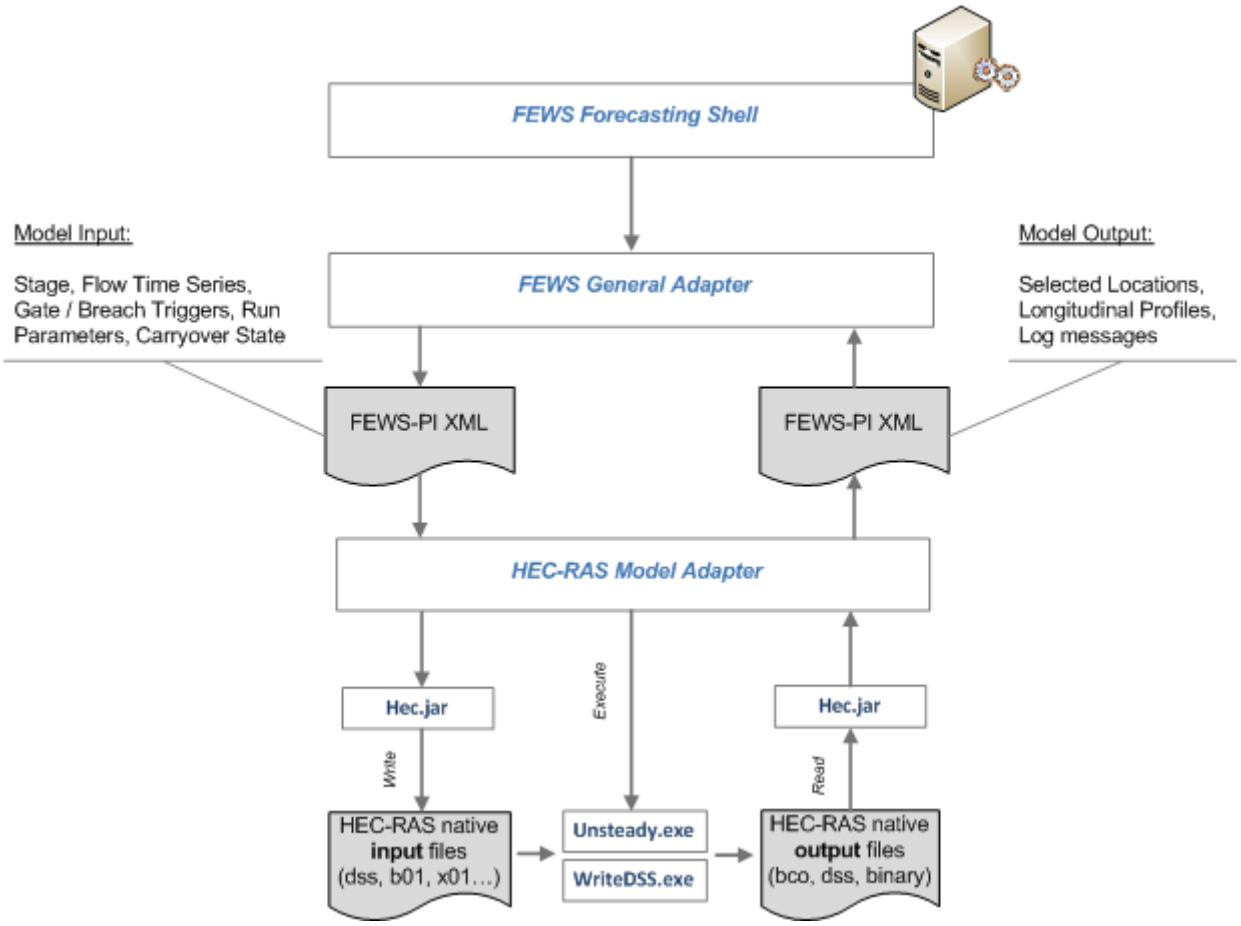

Figure 1: Data flows involved during run of HEC-RAS model FEWS adapter

The FEWS Adapter allows running of HEC-RAS by FEWS. The FEWS Adapter should be considered as a thin communication (software) layer on top the existing HEC-RAS engine. The adapter is tightly connected to the model engine.

2.4 Directory structure

The directory structure of HEC-RAS in CHPS looks as follows:

```
+---bin
|       <FEWS binaries>
\---nerfc sa

    |
    +---Config
    |   +---ColdStateFiles
    |   |      HECRAS CONNECTICTUT UpdateStates Default.zip....... cold state files
    |   |
    |   +---IdMapFiles
    |   |       IdExportHECRAS.xml.......................mapping between ras and fews
    |   |
    |   +---ModuleConfigFiles
    |   |        HECRAS CONNECTICTUT Forecast.xml............ general adapter file
    |   |
    |   +---ModuleParFiles
    |   |         HECRAS_CONNECTICUT_UpdateStates............. parameter file for mods
    |   |
    |   +---ModuleDataSetFiles
```

```
|            HECRAS CONNECTICUT UpdateStates............ initial hecras files

|    +…..
\---Models
    \---hec\hecras
        +---bin..................................... all hec-ras executables
        |       DSSWriter.exe......................... generates binary file
        |       dss_writer
        |       GeomPreprocessor.exe.................. converts geometry files
        |       geo_pre
        |       Steady.exe............................ steady flow simulations
        |       steady
        |       Unsteady.exe.......................... performs unsteady flow
        |       unsteady
        |       DFORMD.DLL
        |       DFORMDD.DLL
        |       MSVCRTD.DLL
        |       libmtsk.so.1
        |       module-adapter-hec-ras.jar............ pre- and pos- adapter
        |       nwsras.jar.......................main library used by the adapter
        |       heclib6-pc.dll
        |       javaHeclib.dll
                libwldelft native.so
        |       castor-0.9.5p.jar......................
         the rest of the files below are FEWS dependencies used by adapter
        |       commons-cli-1.1.jar
        |       Delft_FEWS_castor.jar
        |       Delft_FEWS_schemas.jar
        |       Delft PI.jar
        |       Delft PI castor.jar
        |       Delft_Util.jar
        |       jaxp-api-1_3.jar
        |       jaxp-dom-1-3.jar
        |       jaxp-sax-1 3.jar
        |       jaxp-xalan-1 3.jar
        |       jaxp-xercesImpl-1_3.jar
        |       log4j-1.2.14.jar
        |       serializer.jar
        |       TimeSeriesImport.jar
        |       xalan.jar
        |       xerces-c 2 8.dll
        |       xercesImpl.jar
        |       xml-apis.jar

        |
        \---connecticut......example, could be any river segment
            |    run info.xml......generated by FEWS containing paths, run options
            |
            +---input............. input directory of the adapter (PI timeseries)
            |       inputs.xml
            |
            +---log..............log messages written by the hec-ras adapter
            |       adapter.xml
            |
            +---output...................................contains HEC-RAS output
            |       output.xml
            |
            \---work..................working directory of the adapters, containing
eg.ctfld2ras.b01, .b02, .c02, .f04, .g02, .p01, .p05, prj, .r01, .u01, .x02
```

3 From HEC-RAS GUI to CHPS

In short, the approach to add a HEC-RAS model in CHPS is as follows.

1 Run HEC-RAS model in its GUI.
2 Copy the hecras file from the HEC-RAS work directory to the CHPS environment
3 Make a new General Adapter file in the /Config/ModuleConfigFiles
4 Make a new Coldstate directory for the HEC-RAS model
5 Extend the IdMapFiles with new hecras locations
6 Make a new Parameter File in the /Config/ModuleParFiles (required for run-time mods)
7 Add correct unit conversions
8 HEC-RAS preprocessing and model instability
9 Add new HEC-RAS workflows
10 Register new files and location in the RegionConfigFiles
11 Add a new directory in /Models/hec/hecras (or your preferred file output location).
12 Add files for Interactive Forecast Display
13 Create Filters for time series and longitudinal profiles
14 Add global properties for HEC-RAS model and binaries

The steps will be explained in more detail in the following paragraphs.

3.1 Run HEC-RAS model in the HEC-RAS GUI

As a first step, the HEC-RAS model should be run in its own GUI. Make sure the version is up to date. Be aware that the previous hecras version, 4.01 beta, is replaced by version 4.1. This HEC-RAS version can be downloaded for free at the HEC website. Using HEC-RAS version 4.1 executables in CHPS requires the recent IO library made by RMA. Of particular note is that the new unsteady compute program is NOT compatible with the 4.0.1 beta restart files (*.rst). This means new restart have to be created in the HEC-RAS GUI 4.1, to be used in CHPS when the HEC-RAS 4.1 version is used there.

It is recommended to store the HEC-RAS files in a local directory, separated from the CHPS configuration.

3.1.1 Open HEC-RAS

Open the project file in the work directory, see Figure 2.

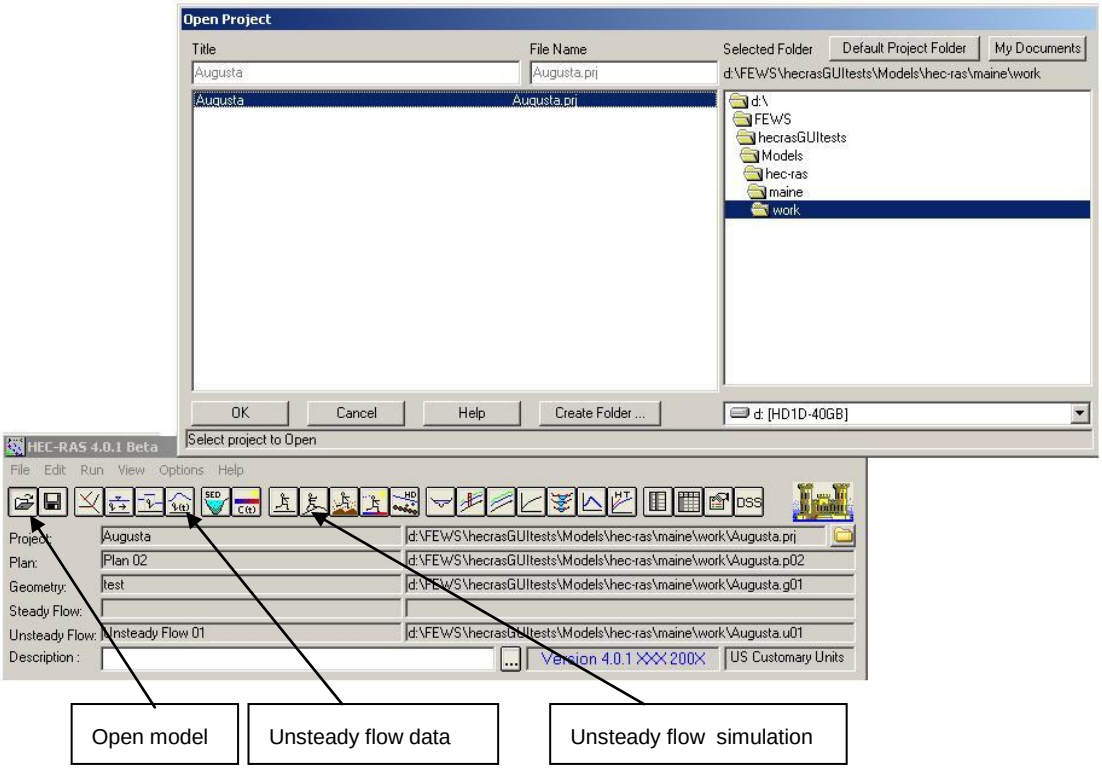

Figure 2: Open HEC-RAS model in the GUI

3.1.2 Initial conditions.

Open the *unsteady flow data* display (see Figure 2) and select the *initial condition* tab. Make sure the model does start with a restart. The name of the restart file is like ctdamlevee.p02.rst. The name of the restart files from HECRAS 4.1 includes date information, but for warm state handling reasons, that part of the name is removed by the HECRAS model adapters after each run. This means that a restart file generated in Windows with a date (e.g. ctdamlevee.p02.29APR2008 2400.rst) should be renamed to something

without a date (e.g. ctdamlevee.p02.rst) for use in Linux. The General Adapter File (Section 3.3) will reference the name without the date.

Note that if you do not have a restart file in Windows, you can first make a run using the "Enter Initial Flow Distribution" option; however, be sure that the last unsteady flow file saved prior to transition to Linux has the option to "Use a Restart File" selected.

Select the *'Boundary Conditions'* tab.

IMPORTANT: Enter the boundary conditions in the tables, not in dss files. Although inconvenient for large models, this is required for the HEC-RAS Adapter to work.

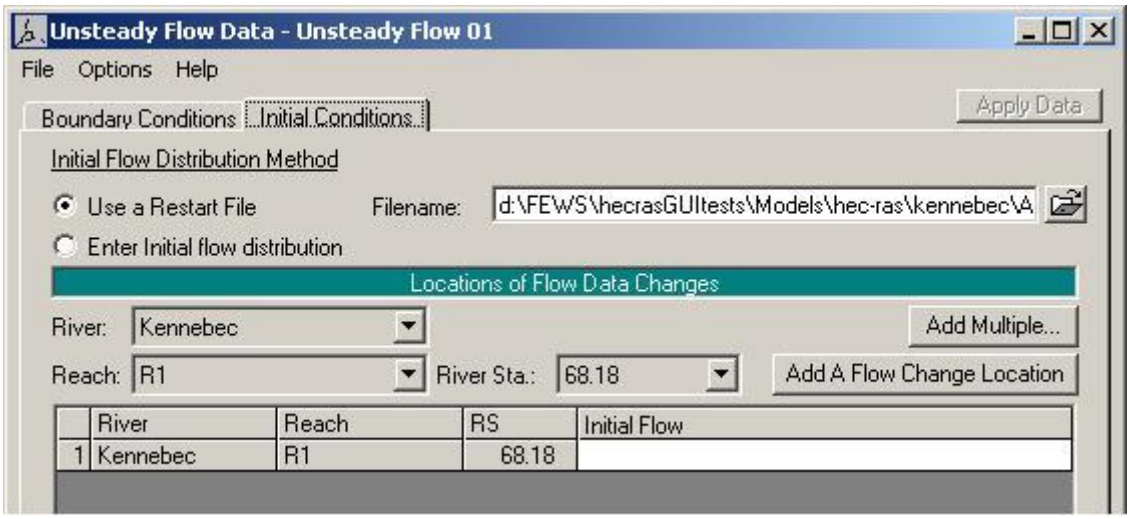

Figure 3: Initial Conditions

3.1.3 Output locations

Open the Unsteady flow simulation display (see Figure 2) and select *'Options/Stage and Flow Output Locations*'. You will see a display that looks like Figure 4.

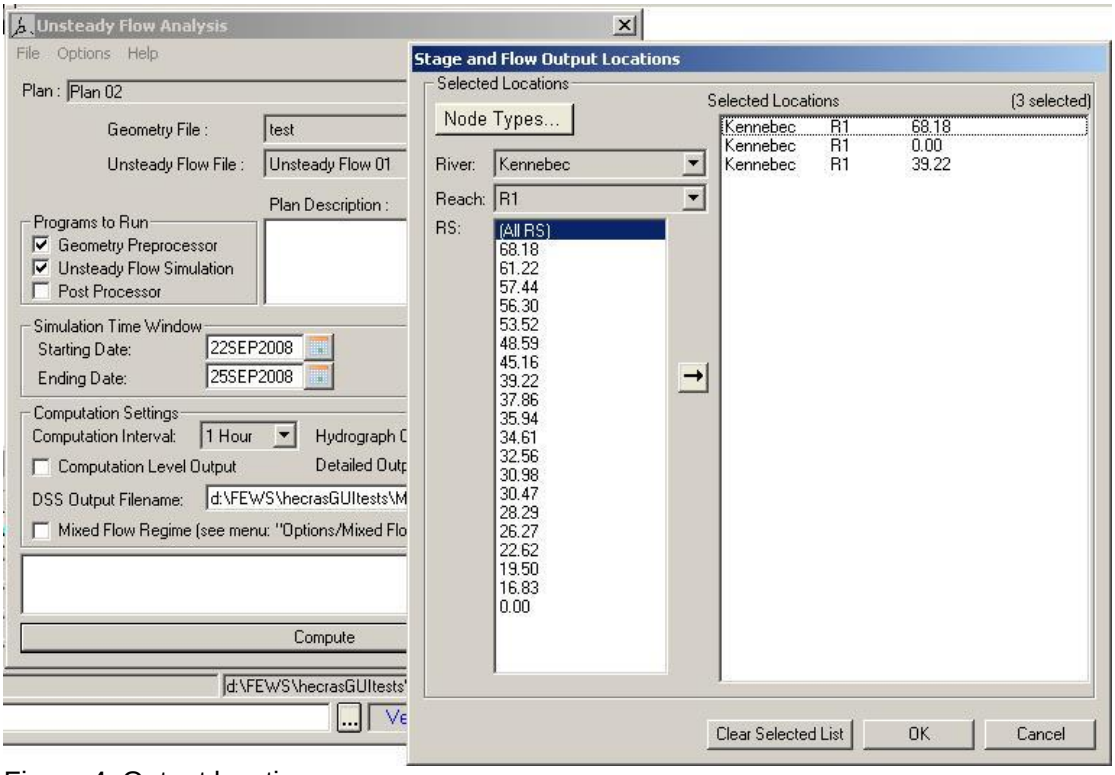

Figure 4: Output locations

Check if the locations in the display are all the desired output locations. HEC-RAS will only write results to those locations (in addition to boundary locations). If not, add more locations and press 'OK' and save the unsteady flow file.

3.1.4 Write restart file

Open the Unsteady flow simulation display (see Figure 2) and select *'Output Options'*. The following display appears.

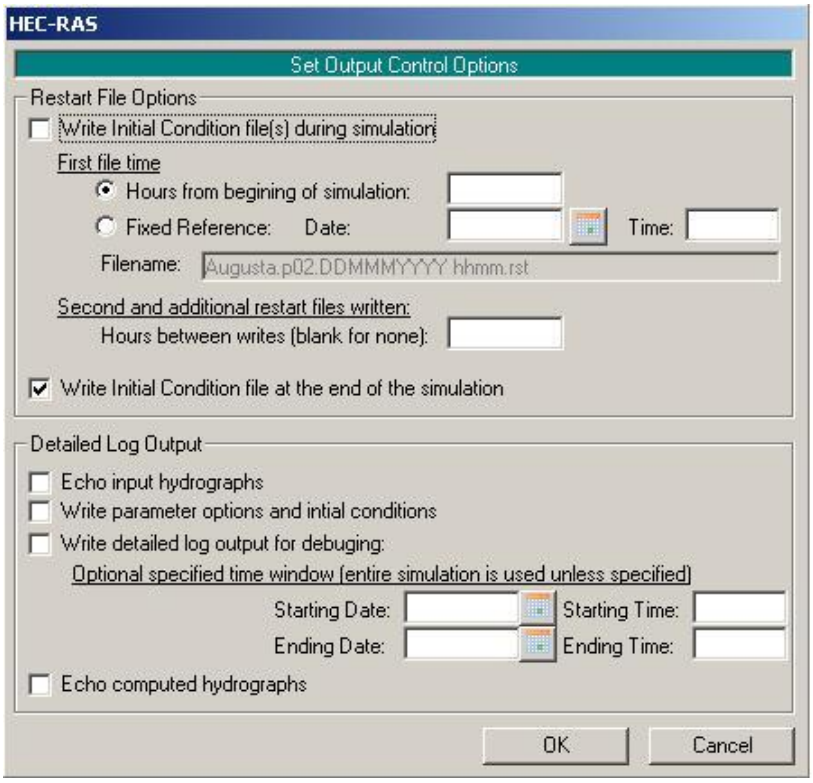

Figure 5: Write restart file

Make sure there is a mark for *'Write Initial Conditions file at the end of the simulation*'. FEWS needs this .rst file for state handling. It needs to be copied to the ColdStateFiles directory (see Paragraph 3.4). FEWS will not use additional initial condition files written during the simulation, so there is no need to choose the option "Write Initial Condition file(s) during simulation."

3.1.5 Run simulation

Open the *Unsteady flow simulation display*, see Figure 2. Check if the *Geometry Preprocessor, Unsteady Flow Simulation* and *Post Processor* are selected, see Figure 6.
The Post Processor is required when variables other than flow or stage that only get written to the flat 'binary' output file ('*.0*') are desired (e.g. velocity, conveyance, etc.). It is recommended to turn on this option in Windows. A CHPS parameter can override this selection once the model is configured on Linux.

Choose a simulation period that covers the boundary data. Choose the desired computational interval and output interval.

Click '*Compute'* to run the model.

Check if no errors occur. If errors exist about missing values, check if the simulation period matches the data in the unsteady flow data boundary conditions.

Save the project.

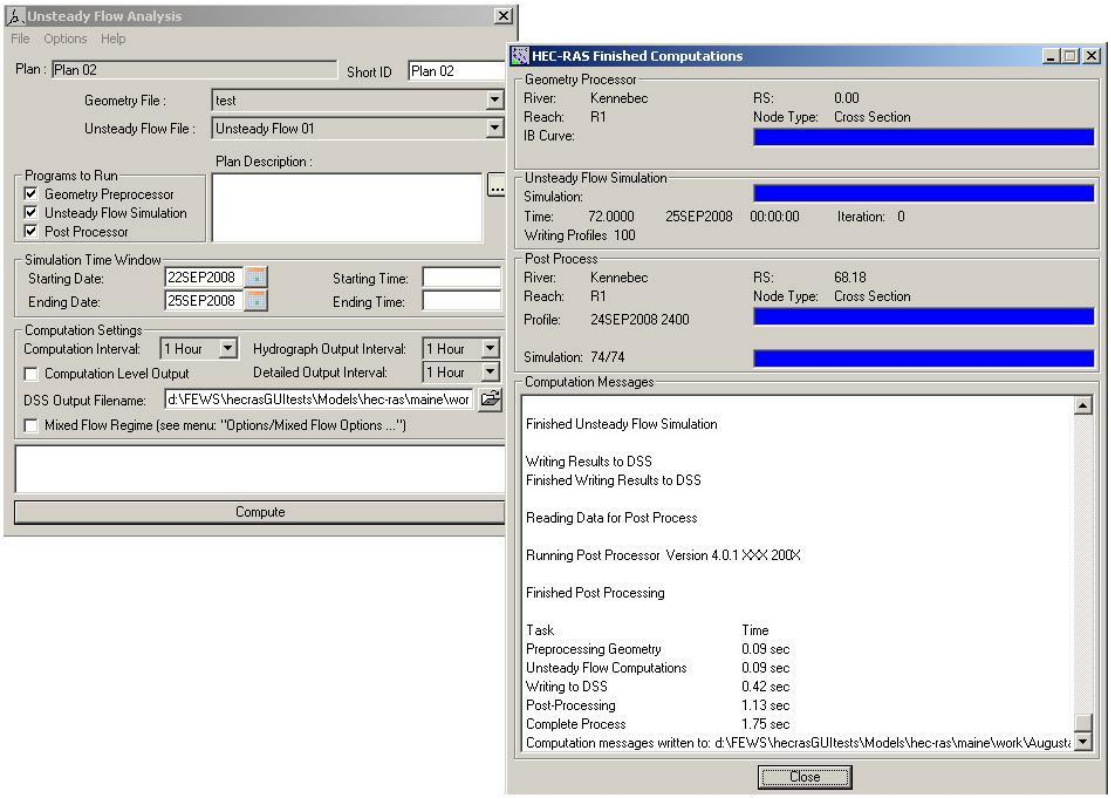

Figure 6: Run Simulation

3.2 Copy the HEC-RAS data to the CHPS environment

The HEC-RAS files in the local work directory must be copied to the CHPS environment. These file are:

- .prj
- .p01
- .g01
- .u01
- .b01
- .x01
- .r01
- .c01

The numbering can be different. Since the HEC-RAS GUI is Windows based, these files are Windows based as well.

Transfer these files to Linux, convert the files form DOS to UNIX file format (applies to ASCII files only), and create a new zip file in the CHPS/Linux environment /Config/ModuleDataSetFiles, eg. HECRAS_KENNEBEC_UpdateStates.zip.

One method for DOS to UNIX conversion is using the 'dos2unix <filename>' command.

To use the command, simply type the command followed by the name of the file you wish to convert, and the name of a file which will contain the converted results. Thus, to convert a Windows file to a Unix file, at the Unix prompt, enter:

dos2unix winfile.txt unixfile.txt.

To convert all files in a directory, use
dos2unix *.

The command 'hexdump –cb <filename>," can be used to check if the file is UNIX or PC. If Windows, you see many /n and /r characters in the file. If the file is UNIX, you see only /n.

IMPORTANT NOTE: If you generate these files using HEC 4.1, you will need to be sure that the *.b01 and *.u01 files reference the same restart file name used in the General Adapter file/ModuleConfigFiles directory (see section 3.12 for discussion of restart file names). **It is recommended to edit the restart file names in *.b01 and *.u01 produced by Windows HECRAS to eliminate the data time stamp after transferring the files to Linux.** For example, 'ctdamlevee.p29.01FEB2008 0100.rst' would be changed to 'ctdamlevee.p29.'

3.3 General Adapter file in the /Config/ModuleConfigFiles

The General Adapter is part of DELFT-FEWS. It is responsible for the data exchange with the models and for executing the models and their adapters, see Figure 7. The General Adapter module can be configured to carry out a sequence of five types of tasks;

- Startup Activities. These activities are run prior to a module run and any export import of data. The activities defined are generally used to remove files from previous runs that may implicate the current run.
- Export Activities. These activities defined all items to be exported through the published interface XML formats to the external module, prior to the module or the module adapters being initialized.
- Execute Activities. The execute activities define the external executables or Java classes to be run. Tracking of diagnostics from these external activities is included in this section.
- Import Activities: These activities define all items to be imported following successful completion of the module run.
- Shutdown Activities. These activities are run following completion of all other activities The activities defined are generally used to remove files no longer required.

The general part administrates the needed files for the general adapter in terms of file names and locations.

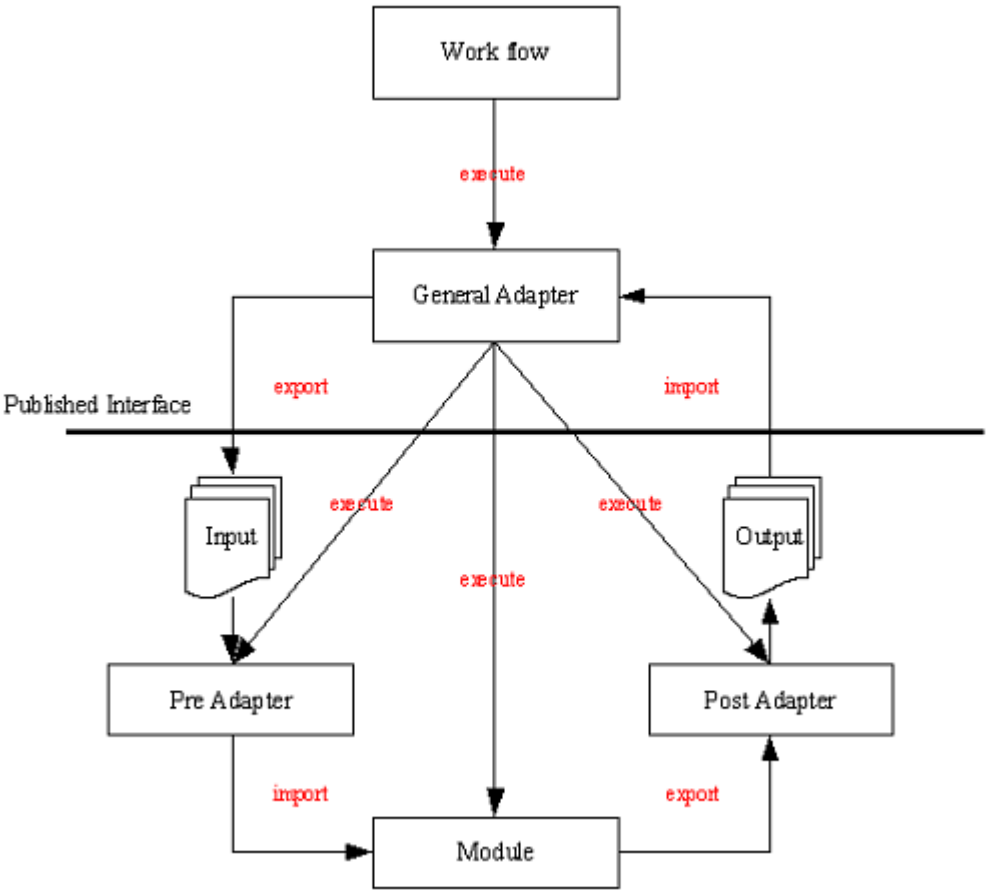

Figure 7: Schematic overview of General Adapter tasks

An example of a general adapter for executing a HEC-RAS model is shown in Appendix 2. This is a UpdateStates file. Also a Forecast file has to be created. The main difference is that the Forecast file does not have an importStateActivity (See Section 3.4.2). Furthermore, all time series types are forecasting instead of historical and use a relative view period with addition *endOverrulable="true*". Another difference is the warm state search period, this should be set to 0 (for a Forecast file) instead of -1 (UpdateStates). The UpdateStates search period is set to -1 because the forecast length for UpdateStates is 0 and FEWS will crash if a state at T0 is used because the simulation time is empty.

By the way, in operational systems, you will hardly see problems for update runs if you have a state search period till T0, because T0 is most of the times ahead of the most recent state in the database. You will get problems if you do the same update run again, if the first run was already succesful and provided as state.

In the example of Appendix 2, during the Startup Activities the “work” (exportDataSetDir) directory is cleaned out prior to each HECRAS run. Data files that were transferred from Wiindows are then copied from the ModuleDataSets directory to “work” and unzipped. The appropriate state (restart) file is then copied to the work directory. If a Cold State is specified in the CHPS GUI, then the restart file is copied from the ColdStates directory (see Section 3.4.1). If a warm state is specified, then CHPS copies the appropriate restart file stored in the CHPS database to the work directory. In either case, the name of the state file in the “work” directory is the same. See Section 3.4.2 for more discussion on State Management.

During the exportTimeSeriesActivity, the appropriate time series for all locations required to run the HECRAS model (specified in IDExportHECRAS.xml) are written to the file “input.xml”. When the HECRAS Adapter is actually executed, time series from input.xml get written to the Boundary Conditions (“*.b01”) file, which is one of the input files read by the HECRAS unsteady executable. The Adapter also updates the starting and ending dates for the run in the HECRAS input files (“*.b01” and “*.p01”).

To execute HECRAS, the CHPS Adapter makes system calls to the HECRAS binaries called “geo_pre” (Geometry Preprocessor), “unsteady” (Unsteady Flow Computational Engine), “dss_writer” (Writes unsteady flow output to DSS files), and “steady”. The “steady” executable is equivalent to selecting the “Post Processor” option in Windows and creates a binary flat file (“*.O01”). The DSS writer only writes stage and flow output to DSS. The “steady” executable creates binary output for many variables that may be of interest. To see a list of available variables and other detailed outputs in the log file, the user may override logging level of the adapter to DEBUG instead of INFO as shown below. Any output variables that the model writes to the “*.O01” file can be consumed by the Adapter and imported back into the CHPS database. The following is the relevant section from the example configuration file.

```
<string key="outputTimeSeriesParametersFilter" value="^STAGE$|^FLOW$"/>
<string key="outputLongtitudionalProfileParametersFilter" value="^STAGE$|Hydr Radius L"/>
<string key="logLevel" value="INFO"/>
<string key="skipBinaryOutput" value="false"/>
```

The skipBinaryOutput option is equivalent to skipping the PostProcessor in Windows. This will improve model performance but won’t allow display of variables other than stage or flow in the CHPS GUI. To get a rough idea of the time savings, a 3 month simulation run took 26 seconds with binary output and only 19 seconds without binary output. Note that the setting of

the “RunPostProcess” option in the HECRAS Plan File does not affect the system call to “steady” (the PostProcessor) implemented by CHPS. Therefore, the skipBinaryOutput variable in CHPS overrides this specification in the Plan file.

The general adapter file is stored in the /Config/ModuleConfigFiles// directory. Steps to create a new general adapter for a HEC-RAS model:

- In general, replace the ‘kennebec’ name for the hecras river name or segment. Each hecras model can have its own place in the /Models/hec/hecras directory, see directory structure Paragraph 2.3.
- In the general setting, make sure your file names and locations are correct and do exist.
- In the exportStateActivity, make sure the name of the state file correspond with the restart file in hecras, eg. Augusta.p02.rst. This file should be located in the /Config/ColdstateFiles// directory. In the filesystem, the restart file must be stored as a zip file with a specific naming convention: HECRAS_KENNEBEC_UpdateStates Default.zip
- Specify the required boundary time series in the exportTimeSeriesActivity. These come either from upstream segments, or from a HECRAS_PreProcessing file, see Paragraph 3.8. Since the location identifiers and parameter names in HEC-RAS are different than in CHPS, these should be mapped in the IdExportHECRAS file, see Paragraph 3.5.
- Make sure the name in the exportDataSetActivity correspond with the name defined in the general setting and in the /Config/ModuleDataSetFiles. This activity copies the hec/ras file to the /Models/hec/ras//work directory (or whichever working directory you specify).
- Make sure the name in the exportParameterActivity correspond with the name in the /Config/ModuleParFiles/. If not present yet, create the parameter file, see Paragraph 3.6. The parameter file holds all HEC-RAS parameters that can be modified in CHPS in the modifier display. For example the computation interval and gate settings.
- Make sure the exportRunFileActivity has the correct HEC-RAS project file name. The run file contains paths to the RAS project file, location of the RAS binary files and list of variables to be written into the output files.
- Make sure the path of the run file as argument in the executeActivity is correct. Special Note for Linux: An environment variable is required. Therefore the following property line is needed when running on Linux:

  ```
  <string key="hecRasEnvironment"
  value="LD_LIBRARY_PATH=$HECRASBINDIR$:$LD_LIBRARY_PATH"/>
  ```

- Add the desired output time series in the importTimeSeriesActivity. Since the location and parameters in HEC-RAS are different than in CHPS, these should be mapped in the IdImportHECRAS file, see Paragraph 3.5.

3.4 HECRAS State Management in CHPS

3.4.1 Coldstate directory for the HEC-RAS model

Add a zip file in the ColdStateFiles// directory, eg. HECRAS_KENNEBEC_UpdateStates Default.zip.

Copy the HEC-RAS restart file to this zip file, eg. Augusta.p02.rst

Example on Linux:
Zip 'HECRAS_KENNEBEC_UpdateStates Default.zip' Augusta.p02.rst

3.4.2 State Management

As specified in Sections 3.1.2 and 3.1.4, the Unsteady Flow file transferred from Windows should be set to read states at the beginning of the run and write/save model states at the end of the run. HECRAS 4.1.0 creates a state file with a date in the name, e.g. Augusta.p02.05FEB2009 1200.rst in the work directory; however, upon completion of the HEC-RAS execution, the CHPS Adapter will make a copy of this file and overwrite the file with the name specified in the General Adapter File (ModuleConfigFiles directory, e.g. Augusta.p02.rst). The importStateActivity specified in the UpdateStates configuration file, reads this renamed file into the FEWS database so that it is available for future forecast runs. Thus, all initial state files have the same name from the viewpoint of HECRAS execution and the dates associated with state files are managed by CHPS/FEWS.

Looking at the stateConfigFile (state.xml here) in the work directory may be helpful in understanding how HECRAS states are being managed by CHPS after each model run. For example, when an UpdateStates run is made including an importStateActivity, the date printed in the state.xml file corresponds to the final date of the HECRAS run, and this is the time this corresponding to the warm state file that gets ingested into CHPS. When a Forecast run is made, which does not include the importStateActivity, the date in the state.xml file corresponds to the date at which the initial state file (warm or cold) was written from the CHPS database to initialize the run.

Another option to check for which date a state is saved in the CHPS database, is using the shortcut F12-1 in the Manual Forecast Display of a CHPS application, namely 'log available states time for selected workflow'. Selecting for example the *HECRAS_LWRCT_UpdateStates* workflow and pressing F12-1 in this display, gives a log info in the log viewer: *INFO - Found state for HECRAS_LWRCT_UpdateStates 14-10-2010 12:00:00*

3.5 IdMapFiles with new HEC-RAS locations

Since the location and parameters in HEC-RAS are different than in CHPS, these should be mapped in the HEC-RAS idMapFiles. The IdExportHECRAS.xml and the IdImportHECRAS.xml. An example of the IdExportHECRAS file is shown below:

```
<map internalLocation="TMVC3" externalLocation="CT RIVER R1/334752.0" internalParameter="QIN"
externalParameter="Flow Hydrograph"/>

<map internalLocation="LYMC3" externalLocation="CT RIVER R1/0.00" internalParameter="STG"
externalParameter="Stage Hydrograph"/>

<map internalLocation="TARC3" externalLocation="CT RIVER R1/34848.00" internalParameter="QIN"
externalParameter="Lateral Inflow Hydrograph"/>

<map internalLocation="structure3" externalLocation="CT River R1/60000" internalParameter="GateOpening"
externalParameter="Gate Openings"/>
```

Add to this file the internal CHPS locations and parameters and map those to the external HEC-RAS locations and parameters.

When specifying location IDs for HECRAS boundary conditions that are inputs to HECRAS (IdExportHECRAS.xml), in most cases, the correct IDs can be constructed from information in the Windows Unsteady Flow Data GUI. For example, the ID for the upstream Flow Hydrograph on the Columbia River in the example of Figure 8 would be “Columbia Lower_1/145.5”.

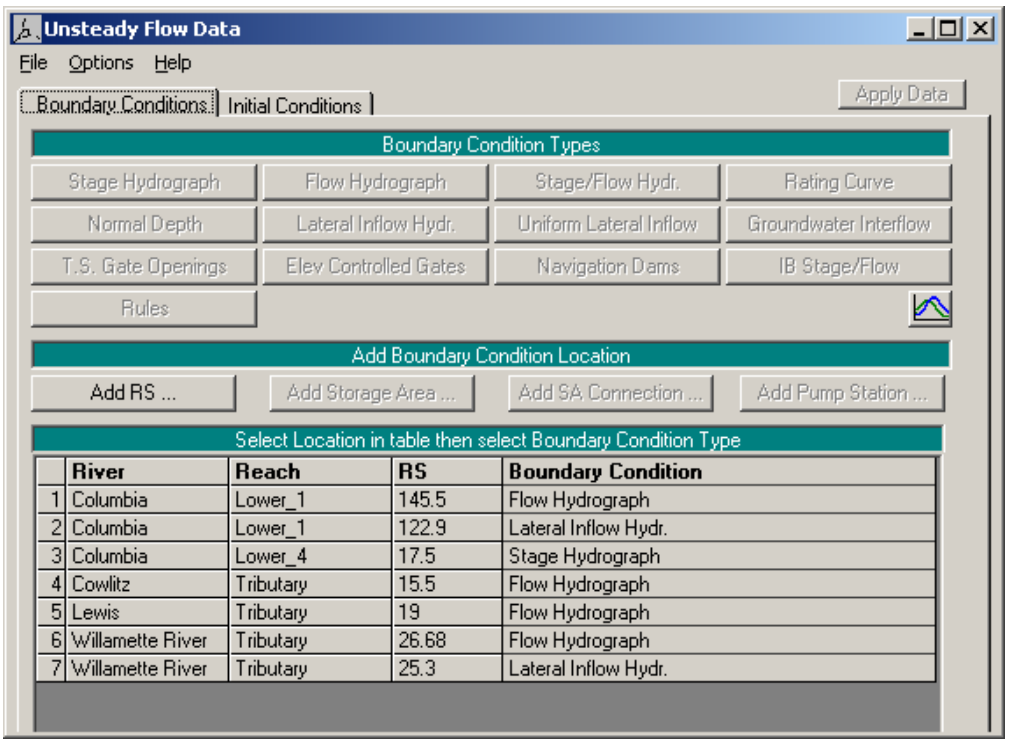

Figure 8. Example of Unsteady Flow Data GUI

However some HECRAS boundary conditions, such as the Uniform Lateral Inflow and Groundwater Inflow are not point inputs, so the ID needs to include both the starting and ending river station. For example, the correct ID might be “RiverName RiverReach/10/1”. The ending station information is not displayed in the main Unsteady Flow GUI, but can be found by double clicking on the boundary condition to get more information or can be found in the Unsteady Flow text file (*.u01) next to the key words “Boundary Location”.

For the case when a Lateral Inflow Hydrograph flows directly to a HEC-RAS Storage Area, the externalLocation name is simply the name of the Storage Area. For example, the following ID mapping can be used to send QIN from point TARC3 to Storage Area “100.”

```
<map internalLocation="TARC3" externalLocation="100" internalParameter="QIN" externalParameter="Lateral Inflow Hydrograph"/>
```

When the HEC-RAS Adapter is run, a printout of the available boundary condition types is specified (in the “Logs” GUI window and in the ./log/preadapter.xml file). Not all of these options require a dynamic time series passed from the CHPS database. Boundary locations requiring time series must be specified properly in the IdMapFile. No IdMapFile entries are required for boundary locations with static parameters.

Boundary Condition Types Requiring Time Series:

- Stage Hydrograph
- Flow Hydrograph
- Stage And Flow Hydrograph (same as “Stage/Flow Hydr” in the HEC-RAS GUI)
- Lateral Inflow Hydrograph
- Uniform Lateral Inflow Hydrograph
- Ground Water Interflow
- Gate Openings (same as “T.S. Gate Openings” in the HEC-RAS GUI)
- Observed Stage and Flow Hydrograph (same as “IB/Stage Flow” in the HEC-RAS GUI)

Note that you must use the exact terms listed above in the IdMapFile, which are sometimes different than those in the HEC-RAS GUI as noted. Missing data are not allowed for all cases except “Observed Stage and Flow Hydrograph” and should be filled in by a pre-processing step. For the case of “Observed Stage and Flow Hydrograph”, at least three observed values are required, but missing data is allowed after that if a rating cuve is defined for the corresponding cross-section. The HEC-RAS model will switch to a rating curve when observed data run out. In this case, data must be missing for all remaining time steps.

Boundary Condition Types not Requiring Time Series

- Rating Curve
- Friction Slope
- Elev Controlled Gate
- Navigation Dam

Output information going back to FEWS (IdImportHECRAS.xml) is normally specified only for point locations and the IDs for these points are available through the HECRAS GUI (e.g. Figure 4) or in the Plan File (*.p01) next to the key word “Stage Flow Hydrograph”.

3.6 Parameter File in the /Config/ModuleParFiles

The parameter file holds all HEC-RAS parameters that can be modified in CHPS in the modifier display, for example the computation interval and gate settings. An example of such a file is given in Appendix 3.

Copy this file to /Config/ModuleParFiles/

This file should be referenced by the exportParameterActivity in the General Adapter File (see Appendix 2).

```
<exportParameterActivity>
        <fileName>params.xml</fileName>
        <moduleInstanceId>HECRAS_KENNEBEC_UpdateStates</moduleInstanceId>
</exportParameterActivity>
```

If the HEC-RAS model has no structures, delete all parameter groups, except the parameter group holding the computational interval parameter.

If the HEC-RAS model has structures, add a parameter group for each structure location and add the HEC-RAS location, eg. CT River R1/220000

Gate parameters depend on the mode selected in the RAS configuration files (gui). Possible modes which can be set in the gui are:

- based on upstream WS (default)
- based on specified reference
- based on difference in stage

The parameters required for each mode are explained in Appendix 3. The others can be disabled or deleted.

3.7 UnitConversionsFiles; add conversions for HEC-RAS

In most cases, the HEC-RAS units are in English units. Since parameters in the CHPS database are in metric units, a correct unit conversion is required. In the general adapter, general settings, two lines can be found about the conversion file, see Appendix 2:

- <exportUnitConversionsId>displayEnglishUnits</exportUnitConversionsId>
- <importUnitConversionsId>ImportEnglishUnits</importUnitConversionsId>

Add the following conversion to the ImportEnglishUnits.xml file in the /Config/UnitConverson directory;

```
<unitConversion>
        <inputUnitType>FEET</inputUnitType>
        <outputUnitType>M</outputUnitType>
        <multiplier>0.3048</multiplier>
        <incrementer>0</incrementer>
</unitConversion>
```

Add the following conversion to the displayEnglishUnits.xml.xml file in the /Config/UnitConverson directory;

```
<unitConversion>
        <inputUnitType>M</inputUnitType>
        <outputUnitType>FEET</outputUnitType>
        <multiplier>3.280839895013</multiplier>
        <incrementer>0</incrementer>
</unitConversion>
```

Note: to check or change the HEC-RAS units in the HEC-RAS GUI, click on 'Options/Unit System in the GUI, see Figure 9. In case it is desired to convert the units in HEC-RAS, click on 'Options/Convert Units

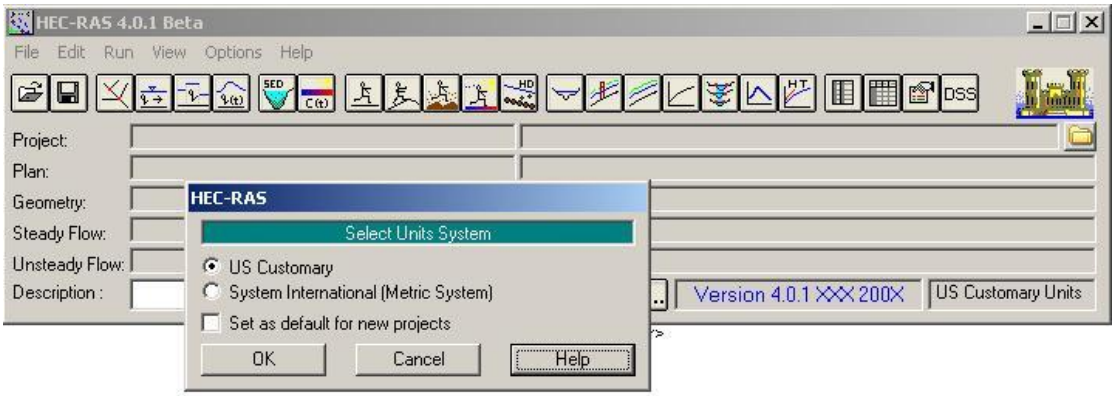

Figure 9: Unit Conversion in HEC-RAS

3.8 HEC-RAS preprocess file and model instability

Since HEC-RAS is a hydrodynamic model, instability problems may occur, for example due to too low flows or too big changes in boundary conditions. To avoid these, it is recommended to add a preprocess step on the boundary conditions prior to running the model.

3.8.1 Minimum flow

Appendix 4 holds an example of a preprocess file. It takes care that the boundary data is interpolated in case of missing data and a minimum flow is guaranteed.

In case this file is desired, add the boundary time series to this file and choose a minimum flow for the model. Alternatively, minimum flow can also be specified internally within HEC-RAS; however, any missing data must be filled in by CHPS with the exception of internal boundaries with rating curves as discussed in Section 3.5.

3.8.2 Burn-in profile

Model instability can occur when starting with a cold start in combination with a boundary flow which is far too high. To avoid this, a burn-in profile can be added to the general adapter, after the general setting. An example is given below

```
<burnInProfile>
        <length unit="hour" multiplier="48"/>
        <timeSeries>
                <parameterId>QIN</parameterId>
                <locationId>TMVC3</locationId>
                <initialValue>600</initialValue>
        </timeSeries>
        <timeSeries>
                <parameterId>SSTG</parameterId>
                <locationId>LYMC3</locationId>
                <initialValue>0.50</initialValue>
        </timeSeries>
</burnInProfile>
```

The length unit defines the period in which the initial value blends to the simulated values.

Note: In operational settings, a cold start never occurs, since there is a so-called warm state from recent runs. This ensures the state and the boundary flows are in balance.

3.9 HEC-RAS workflows

To run the new HEC-RAS general adapter, it should be listed in the segment workflows in the /Config/WorkflowFiles directory. These workflows are:

- <segmentname>_Flow_UpdateStates.xml
- <segmentname>_Flow_Forecast.xml

Example for the UpdateStates workflow for Kennebec:

```
<activity>
        <runIndependent>true</runIndependent>
        <moduleInstanceId>HECRAS_KENNEBEC_Preprocessing_UpdateStates</moduleInstanceId>
</activity>
<activity>
        <runIndependent>true</runIndependent>
        <moduleInstanceId>HECRAS_KENNEBEC_UpdateStates</moduleInstanceId>
</activity>
```

For testing reasons, it can be useful to make a temporary workflow with only the HEC-RAS module instance, instead of running the whole upstream chain of models. However, the upstream part should run at least one time, since the boundary conditions for the HEC-RAS model must be created.

3.10 Register new files and location in the RegionConfigFiles

Most of the new created files and internal locations need to be registered in ModuleInstanceDescriptorFile.xml, WorkflowDescriptorFile.xml, and Locations.xml. These files can be found in the /Config/RegionConfigFiles.

3.11 Add a new directory in /Models

The directory structure for the /Models/hec/hecras is shown in Section 2.4. The HEC-RAS executables, run_info.xml, input FEWS-PI time series, log messages written by the adapter, HEC-RAS output converted from binary and dss output files and work directory of the adapter are located here.

Add a new directory in /Models/hec/hecras, eg. /Models/hec/hecras/kennebec.
Create four new folders in this directory:

- input
- log
- output
- work

The Adapter automatically creates these folders if they do not exist.

3.12 Topology files for Interactive Forecast Display

To make the HEC-RAS run part of the Interactive Forecast Display (IFD), the topology file may need to be extended with a new node id for the new segment. Only if the configuration of the topology file (and the Parameter file) is correct, can HEC-RAS parameters be modified in the modifier displays. An example of a topology file for one forecast group is shown below.

```
<nodes id="RussianNapa" name="RussianNapa">
        <workflowId>RussianNapa_Forecast</workflowId>
        <node id="NVRC1" name="NAVARRO - NAVARRO">
                <workflowId>NVRC1_Forecast</workflowId>
        </node>
        <node id="GRCC1" name="GARCIA R - PT ARENA">
                <workflowId>GRCC1_Forecast</workflowId>
        </node>
        <node id="LAMC1" name="RUSSIAN - LK MENDOCI">
                <workflowId>LAMC1_Forecast</workflowId>
        </node>
        <node id="HOPC1" name="RUSSIAN - HOPLAND">
                <previousNodeId>LAMC1</previousNodeId>
                <workflowId>HOPC1_Forecast</workflowId>
        </node>
        <node id="CDLC1" name="RUSSIAN - CLOVERDALE">
                <previousNodeId>HOPC1</previousNodeId>
                <workflowId>CDLC1_Forecast</workflowId>
        </node>
        <node id="HEAC1" name="RUSSIAN - HEALDSBURG">
                <previousNodeId>CDLC1</previousNodeId>
                <workflowId>HEAC1_Forecast</workflowId>
        </node>
        <node id="WSDC1" name="DRY CK - WARM SPRGS">
                <workflowId>WSDC1_Forecast</workflowId>
        </node>
        <node id="GUEC1" name="RUSSIAN-GUERNEVILLE">
                <previousNodeId>HEAC1</previousNodeId>
                <previousNodeId>WSDC1</previousNodeId>
                <workflowId>GUEC1_Forecast</workflowId>
        </node>
        <node id="SHEC1" name="NAPA - ST HELENA">
                <workflowId>SHEC1_Forecast</workflowId>
        </node>
        <node id="APCC1" name="NAPA - NAPA">
                <previousNodeId>SHEC1</previousNodeId>
                <workflowId>APCC1_Forecast</workflowId>
        </node>
        <mapExtentId>CNRFC_RussianNapa</mapExtentId>
    </nodes>
```

Important is that the node Id corresponds with the segment name in the /Config/ModuleParFiles directory!

In most cases, the new HEC-RAS moduleInstance activity has already been added to the _Flow_Forecast workflow, which is part of the _Forecast workflow. In that case no adjustments of the topology file needs to be made.

However, for testing reasons it is sometimes useful to create a separate node for only the HEC-RAS run. The new node Id can look like:

```
<nodes id="Connecticut" name="Connecticut">
        <workflowId>Connecticut_Forecast</workflowId>
        <node id="hecras2" name="hecras2">
                <locationId>HFDC3</locationId>
                <workflowId>TEST_HECRAS2</workflowId>
        </node>
        <mapExtentId>Connecticut River</mapExtentId>
</nodes>
```

In this case, the parameter file, as referenced in the general adapter file TEST_HECRAS2, should be placed in the temporary /Config/ModuleParFiles/hecras2 directory

3.13 ModifierTypes

HEC-RAS parameter mods can only be used if they are defined in the /Config/RegionConfigFiles/ModifierTypes.xml. Add the following lines:

```
<moduleParameterModifier id="HECRAS" name="HECRAS">
        <defaultValidTime/>
</moduleParameterModifier>
```

Note that the option to make modifications will not be available unless a successful run is completed. Figure 10 shows an example of the Modifier interface for HEC-RAS parameters.

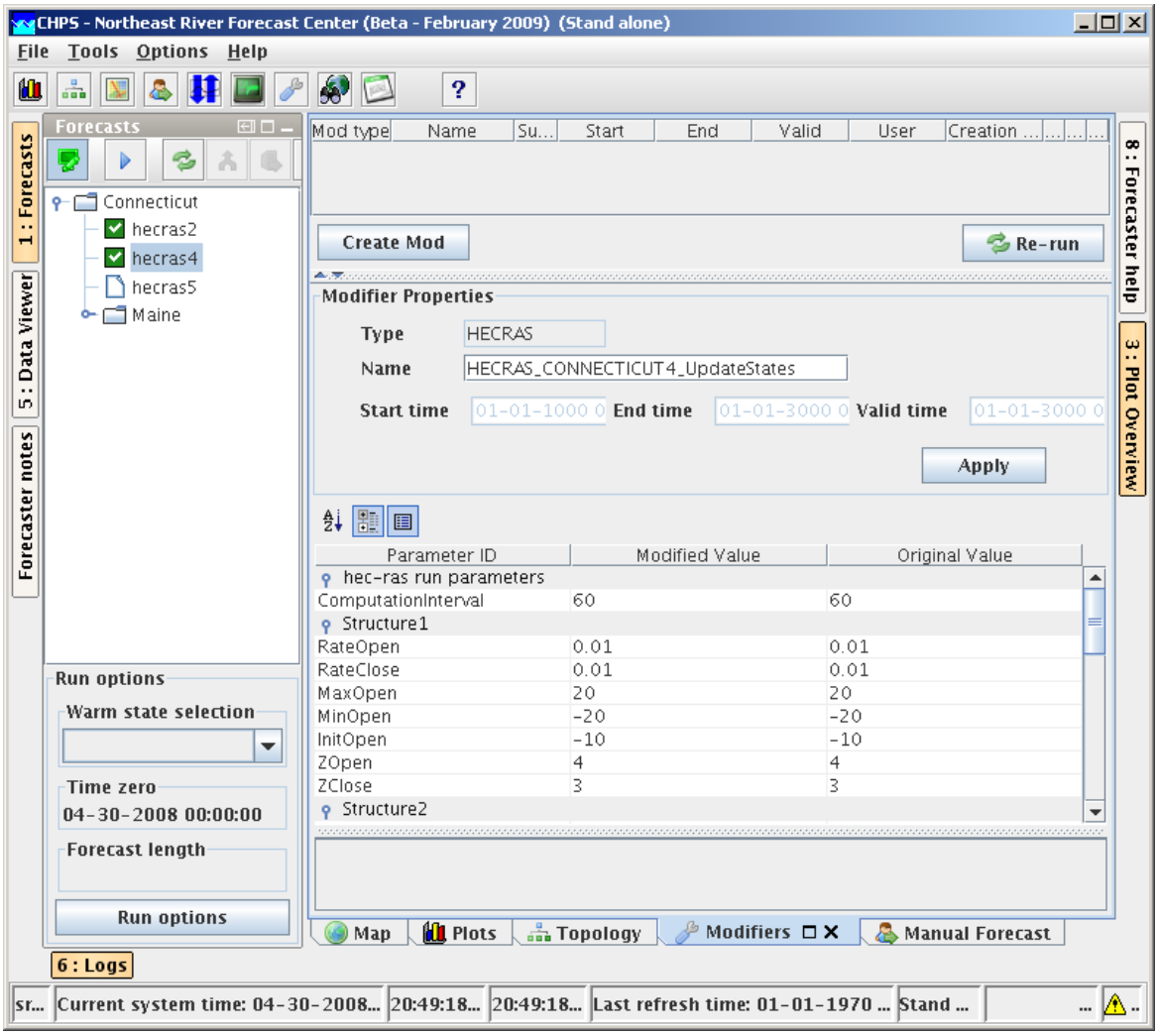

Figure 10. Example modifier interface showing selected HEC-RAS parameters.

3.14 Filters

The input and output timeseries can be filtered from the CHPS database by creating filters. An example of an input and output filter for a Kennebec model can be found in Appendix 5. This can be added to the Filters.xml in the /Config/RegionConfigFiles. The time series can also be added to de DisplayGroups in the /Config/SystemConfigFiles. It is the user's choice how to organize it.

For testing new HEC-RAS models in CHPS, it is useful to create a filter as in Appendix 5. Other tools for viewing data, especially in the test phase, are the Database Viewer and the Workflow Navigator.

3.15 Longitudinal Profiles

In September 2011, functionality was added to the HECRAS model adapter to parse longitudinal profiles for one or multiple reaches from the HECRAS output to CHPS. These profiles can be shown in the Data Viewer or in the predefined shortcuts (DisplayGroups), see Figure 11. The profile can be moved in time using the animation options in the toolbar of the Plots Display.

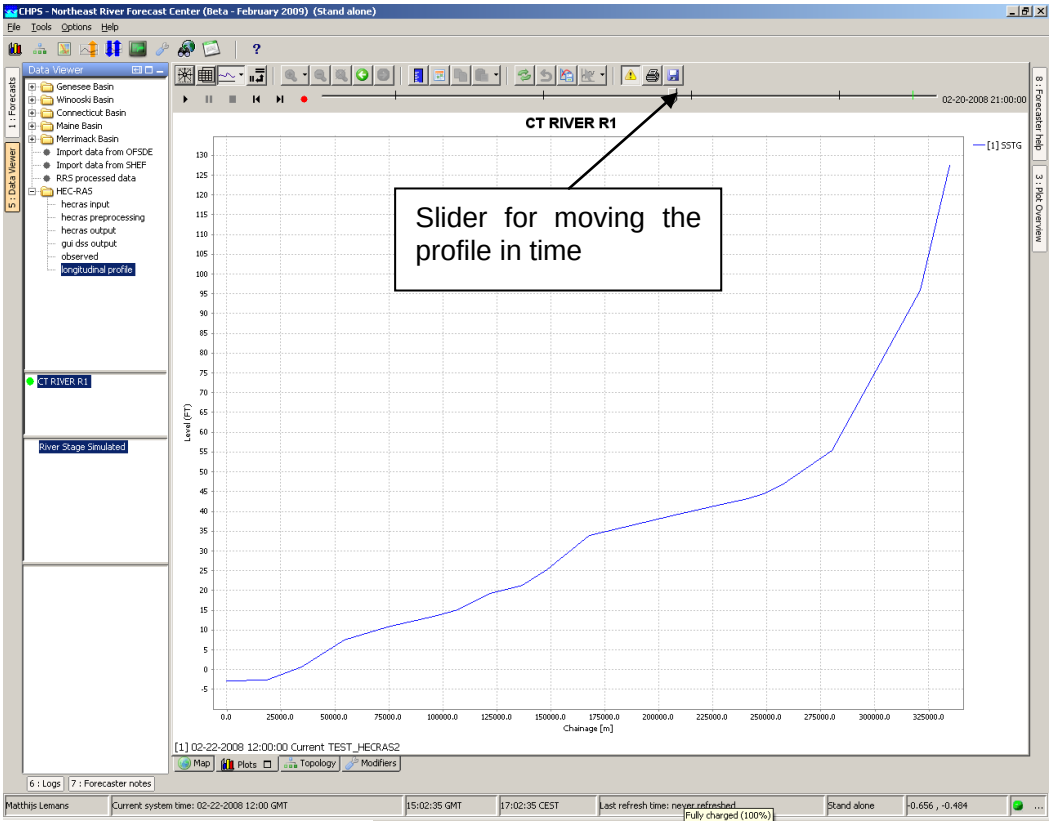

Figure 11: Longitudinal profile

The required configuration additions are:

3.15.1 Additions to General Adapter

Two additions are required. The first includes two extra string keys as property in the exportRunFileActivity. The second key is optional and can be used to filter specific time series.

- *<string key="outputLongtitudionalProfileFile" value="%ROOT_DIR%/connecticut/output/**output_profiles.xml**"/>*
- *<string key="outputLongtitudionalProfileParametersFilter" value="^LOCATION-FLOW$|^LOCATION-ELEV$"/>*

The second addition is an extra importTimeseriesActivity, reading an output_profiles.xml file produced by the model adapter:

Also, the reach location CT RIVER R1 should be registred in the Locations.xml file, with dummy coordinates.

```
<importTimeSeriesActivity>
        <importFile>%ROOT_DIR%/output/output_profiles.xml</importFile>
        <timeSeriesSets>
                <timeSeriesSet>
                        <moduleInstanceId>HECRAS_CONNECTICUT2_Forecast</moduleInstanceId>
                        <valueType>longitudinalprofile</valueType>
                        <parameterId>LOCATION-ELEV</parameterId>
                        <locationId> CT RIVER R1</locationId>
                        <timeSeriesType>simulated forecasting</timeSeriesType>
                        <timeStep unit="hour" multiplier="1"/>
                        <readWriteMode>add originals</readWriteMode>
                </timeSeriesSet>
        </timeSeriesSets>
</importTimeSeriesActivity>
```

A limitation of the current output_profiles.xml file is that it does not contain any information on units. Therefore, the units of the variable used to import the data dictates the units that are assumed. For example, many RFCs define the <unit> for SSTG (simulated stage) as M; therefore, if the parameterId above is set to SSTG, the values in output_profiles.xml will be assumed to be in M. For HEC-RAS applications in the US, it is common to use FT, therefore, the units for elevation/stage values in output_profiles.xml will often be in FT. For this example, a parameter called LOCATION-ELEV has been defined in RegionConfigFiles/Parameters.xml to allow importing values as FT, e.g.:

```
<parameter id="LOCATION-ELEV" name="River Stage Simulated Profile">
        <shortName>LELEV</shortName>
</parameter>
```

3.15.2 Branches.xml file

CHPS needs this file in the RegionConfigFiles to be able to store the profile in its database. The branch Id should correspond with the reach location in the General Adapter file. Multiple branches can be configured in this file.

More information about Branches in FEWS can be found at:
https://publicwiki.deltares.nl/display/FEWSDOC/05+Branches

Example data in a the branches.xml file:

```
<?xml version="1.0" encoding="UTF-8"?>
<branches 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/branches.xsd"
version="1.1">
        <geoDatum>WGS 1984</geoDatum>
<branch id="CT RIVER R1">
                <branchName>CT RIVER R1</branchName>
                <startChainage>0</startChainage>
                <endChainage>334752.0</endChainage>
                <pt chainage="0.0" label="CT RIVER R1_0.000000"/>
                <pt chainage="18480.0" label="CT RIVER R1_18480.000000"/>
                <pt chainage="34848.0" label="CT RIVER R1_34848.000000"/>
                <pt chainage="54384.0" label="CT RIVER R1_54384.000000"/>
                <pt chainage="74448.0" label="CT RIVER R1_74448.000000"/>
                <pt chainage="97679.976563" label="CT RIVER R1_97679.976563"/>
                <pt chainage="107184.0" label="CT RIVER R1_107184.000000"/>
                <pt chainage="121968.0" label="CT RIVER R1_121968.000000"/>
```

```
            <pt chainage="136752.0" label="CT RIVER R1_136752.000000"/>
            <pt chainage="148368.0" label="CT RIVER R1_148368.000000"/>
            <pt chainage="167904.0" label="CT RIVER R1_167904.000000"/>
            <pt chainage="209088.0" label="CT RIVER R1_209088.000000"/>
            <pt chainage="229152.0" label="CT RIVER R1_229152.000000“/>
            <pt chainage="241296.0" label="CT RIVER R1_241296.000000"/>
            <pt chainage="248688.0" label="CT RIVER R1_248688.000000"/>
            <pt chainage="257136.0" label="CT RIVER R1_257136.000000"/>
            <pt chainage="280368.0" label="CT RIVER R1_280368.000000"/>
            <pt chainage="321024.0" label="CT RIVER R1_321024.000000"/>
            <pt chainage="334752.0" label="CT RIVER R1_334752.000000"/>
        </branch>
</branches>
```

The chainages and label name can be derived from the DSS output file (in Windows) by selecting one of the LOCATION-ELEV time series in the DSS viewer and copying the chainages from the table output to a spreadsheet. See Figure 12. The label name is constructed by added the reach name + underscore character+ chainage, eg. CT RIVER R1_18480.000000.

KEY POINTS:
(1) The amount of decimals should be equal to the label name in the output_profiles.xml, otherwise CHPS cannot import the time series. In the September 2011 version of the Adapter, the number of decimal places has been hardcoded to 6. To see the 6 decimal places you need in the DSS Viewer, select Options → Number of Decimal Places under the DSS Plot. Use of a large number of decimal places is an attempt to ensure that no duplicate point locations exist. Testing showed some models can have cross-sections that are very close together.

(2) The FEWS Branches documentation indicates that the startChainage must be numerically greater than the endChainage and that each ‘pt’ must lie in-between these values. This is opposite from the tabular display shown in Figure 12. Therefore, the chainage should be reversed and entered from lowest to highest in the Branches.xml file as illustrated above.

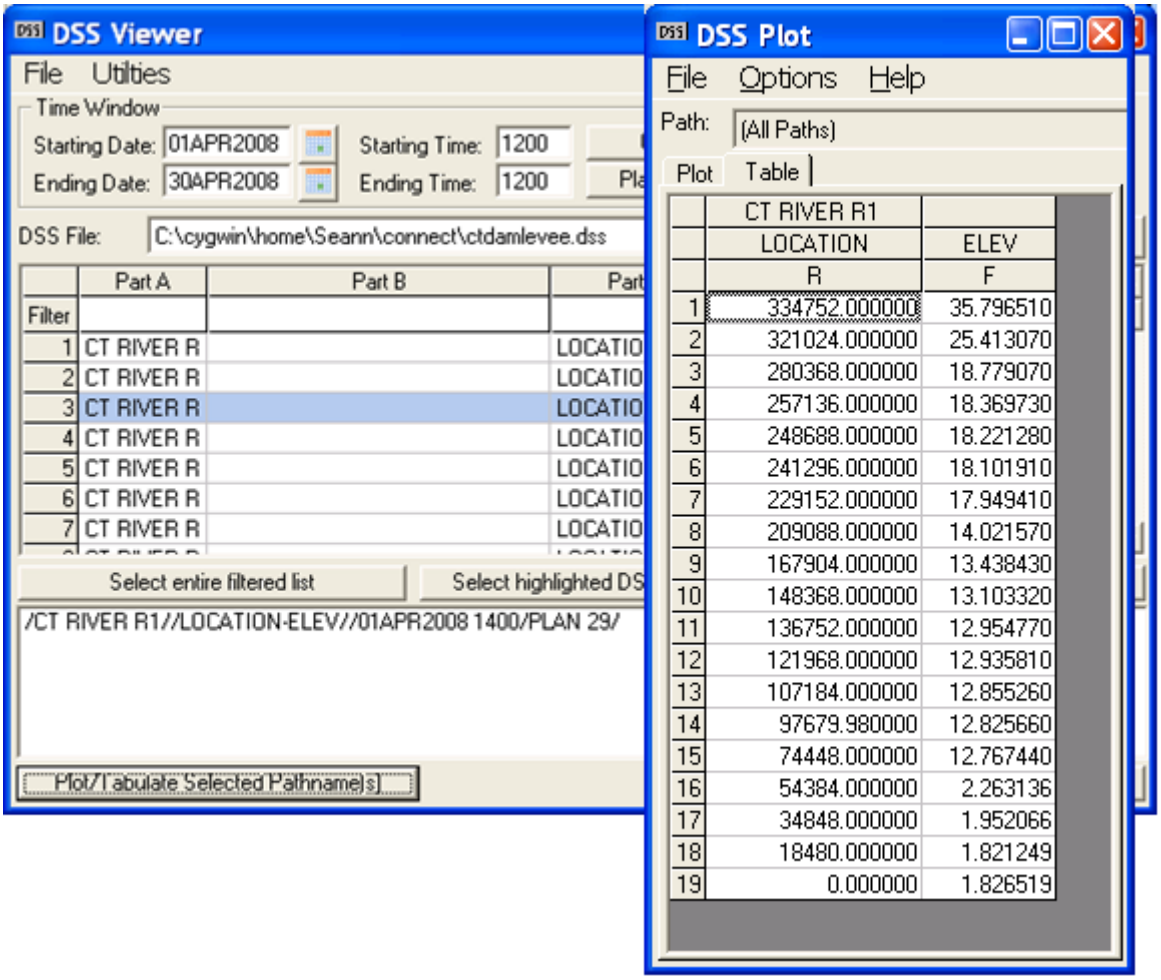

Figure 12: Chainages in HECRAS DSS file

3.15.3 Additions to IdImportHECRAS.xml

Additions to the IdImportHECRAS.xml IdMap file are required to map the HECRAS locations and parameters with the ones in CHPS:

<map internalLocation="CT RIVER R1" externalLocation="CT River R1" internalParameter="LOCATION-ELEV" externalParameter="LOCATION-ELEV"/>

IMPORTANT: The external location is mixed case here because it matches what is written for the location in output_profiles.xml. The internalLocation (FEWS/CHPS) is upper case to match the chainage information written to output_profiles.xml.

3.15.4 Addtitions to Filters.xml or DisplayGroups.xml

Once imported in CHPS, the profile can be shown by configuring a longitudinal time series set in either the Filters.xml or the DisplayGroups.xml. Make sure the valueType is longitudinalprofile and the locationId is equal to the BranchId.

```
<filter id="longitudinal_profile" name="longitudinal profile">
        <mapExtentId>Connecticut River</mapExtentId>
        <timeSeriesSet>
                <moduleInstanceId>HECRAS_CONNECTICUT2_Forecast</moduleInstanceId>
                <valueType>longitudinalprofile</valueType>
```

```
            <parameterId>LOCATION-ELEV</parameterId>
            <locationId>CT RIVER R1</locationId>
            <timeSeriesType>simulated forecasting</timeSeriesType>
            <timeStep unit="hour" multiplier="1"/>
            <readWriteMode>read complete forecast</readWriteMode>
        </timeSeriesSet>
</filter>
```

3.16 Add global properties for hecras model and binaries

The $HECRASBINDIR$ property is defined in the global.properties at the same level of the Config and Models directory:

HECRASMODELDIR=%REGION_HOME%/Models/hec/hecras
HECRASBINDIR=$HECRASMODELDIR$/bin

The General Adapter offers flexibility on where HEC-RAS working files are written (the HECRASMODELDIR). User's may not wish to not to place the executables under the HECRASMODELDIR location as described above for performance reasons (see next Section).

3.17 HEC-RAS Model Performance when run from CHPS on Linux

Numerous factors affect the performance of external models when run from CHPS. There are three factors that are specific to HEC-RAS.

- As discussed in Section 3.3, turning off the post-processor can improve performance.
- Specifying that HEC-RAS output be written to a local disk (e.g. via the HECRASMODELDIR variable) rather than a networked disk can improve performance. The HECRAS Linux executable called 'dss_writer', which is run each time a HEC-RAS model is executed from CHPS seems to be particularly sensitive to this factor.
- As of HEC-RAS 4.1.0 Linux executables, tests indicate the 'dss_writer' tends to run much slower on 64 bit machines than on 32 bit machines. This issue is still being investigated.

Appendix 1: Description of HEC-RAS data files

Table 1 List of files to be read and written by adapter

Extension	Description	pre-adapter input	pre-adapter output	post-adapter input	Post-adapter output
.prj	*project file*	✓		✓	
.p01	*plan files*	✓		✓	
.g01	*geometry files*	✓		✓	
.f01	*flow files*				
.u01	*unsteady flow files*				
.b01	*unsteady run files*	✓	✓	✓	
.x01	*input file for geometry preprocessor*				
.r01	*steady run file*				
.O01	*binary output file*			✓	
.bco	*model log file*			✓	
.c01	output of geometry preprocessor				
.dss	input / output files			✓	
.hyd01	input file for geometry preprocessor				

Appendix 2: General adapter file


```
<?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://fews.wldelft.nl/schemas/version1.0/generalAdapterRun.xsd">
    <general>
        <description>hecras Model for Kennebec River</description>
        <rootDir>$HECRASMODELDIR$/kennebec</rootDir>
        <workDir>%ROOT_DIR%/work</workDir>
        <exportDir>%ROOT_DIR%/input</exportDir>
        <exportDataSetDir>%ROOT_DIR%/work</exportDataSetDir>
        <exportIdMap>IdExportHECRAS</exportIdMap>
        <exportUnitConversionsId>ExportHECRAS</exportUnitConversionsId>
        <importDir>%ROOT_DIR%output</importDir>
        <importIdMap>IdImportHECRAS</importIdMap>
        <importUnitConversionsId>ImportHECRAS</importUnitConversionsId>
        <dumpFileDir>$GA_DUMPFILEDIR$</dumpFileDir>
        <dumpDir>%ROOT_DIR%</dumpDir>
        <diagnosticFile>%ROOT_DIR%/log/preadapter.xml</diagnosticFile>
    </general>
    <activities>
        <startUpActivities>
            <purgeActivity>
                <filter>%ROOT_DIR%/log/*.*</filter>
            </purgeActivity>
            <purgeActivity>
                <filter>%ROOT_DIR%/input/*.*</filter>
            </purgeActivity>
            <purgeActivity>
                <filter>%ROOT_DIR%/output/*.*</filter>
            </purgeActivity>
            <purgeActivity>
                <filter>%ROOT_DIR%/work/*.*</filter>
            </purgeActivity>
        </startUpActivities>
        <exportActivities>
            <exportStateActivity>
                <moduleInstanceId>HECRAS_KENNEBEC_UpdateStates</moduleInstanceId>
                <stateExportDir>%ROOT_DIR%/work</stateExportDir>
                <stateConfigFile>%ROOT_DIR%/work/state.xml</stateConfigFile>
                <stateLocations type="file">
                    <stateLocation>
                        <readLocation>Augusta.p02.rst</readLocation>
                        <writeLocation>Augusta.p02.rst</writeLocation>
                    </stateLocation>
                </stateLocations>
                <stateSelection>
                    <warmState>
```

```
                                        <stateSearchPeriod unit="day" start="-10" end="-1"/>
                                    </warmState>
                                </stateSelection>
                            </exportStateActivity>
                            <exportTimeSeriesActivity>
                                <exportFile>%ROOT_DIR%/input/input.xml</exportFile>
                                <timeSeriesSets>
                                    <timeSeriesSet>

<moduleInstanceId>HECRAS_KENNEBEC_Preprocessing_UpdateStates</moduleInstanceId>
                                        <valueType>scalar</valueType>
                                        <parameterId>QINE</parameterId>
                                        <locationId>SIDM1ME</locationId>
                                        <timeSeriesType>simulated historical</timeSeriesType>
                                        <timeStep unit="hour" multiplier="1"/>
                                        <relativeViewPeriod unit="hour" end="0"/>
                                        <readWriteMode>read only</readWriteMode>
                                    </timeSeriesSet>
                                    <timeSeriesSet>

<moduleInstanceId>HECRAS_KENNEBEC_Preprocessing_UpdateStates</moduleInstanceId>
                                        <valueType>scalar</valueType>
                                        <parameterId>STID</parameterId>
                                        <locationId>CASM1ME</locationId>
                                        <timeSeriesType>simulated historical</timeSeriesType>
                                        <timeStep unit="hour" multiplier="1"/>
                                        <relativeViewPeriod unit="hour" end="0"/>
                                        <readWriteMode>read only</readWriteMode>
                                    </timeSeriesSet>
                                </timeSeriesSets>
                            </exportTimeSeriesActivity>
                            <exportDataSetActivity>
                                <moduleInstanceId>HECRAS_KENNEBEC_UpdateStates</moduleInstanceId>
                            </exportDataSetActivity>
                            <exportParameterActivity>
                                <fileName>params.xml</fileName>
                                <moduleInstanceId>HECRAS_KENNEBEC_UpdateStates</moduleInstanceId>
                            </exportParameterActivity>
                            <exportRunFileActivity>
                                <exportFile>%ROOT_DIR%/run_info.xml</exportFile>
                                <properties>
<string key="hecRasProjectFile" value="%ROOT_DIR%/work/Augusta.prj"/>
<string key="hecRasBinDirectory" value="$HECRASBINDIR$"/>
<string key="outputTimeSeriesParametersFilter" value="^STAGE$|^FLOW$"/>
<string key="outputLongtitudionalProfileParametersFilter" value="^STAGE$|Hydr Radius L"/>
<string key="logLevel" value="INFO"/>
<string key="skipBinaryOutput" value="false"/>
<string key="hecRasEnvironment" value="LD_LIBRARY_PATH=$HECRASBINDIR$:$LD_LIBRARY_PATH"/>
                                </properties>
                            </exportRunFileActivity>
                        </exportActivities>
```

```
        <executeActivities>
            <executeActivity>
                <command>
                    <className>nl.wldelft.fews.adapter.hec.HecRasAdapter</className>
                    <binDir>$HECRASBINDIR$</binDir>
                </command>
                <arguments>
                    <argument>%ROOT_DIR%/run_info.xml</argument>
                </arguments>
                <timeOut>1500000</timeOut>
            </executeActivity>
        </executeActivities>
        <importActivities>
            <importStateActivity>
                <stateConfigFile>%ROOT_DIR%/work/state.xml</stateConfigFile>
                <synchLevel>20</synchLevel>
            </importStateActivity>
            <importTimeSeriesActivity>
                <importFile>%ROOT_DIR%/output/output.xml</importFile>
                <timeSeriesSets>
                    <timeSeriesSet>

    <moduleInstanceId>HECRAS_KENNEBEC_UpdateStates</moduleInstanceId>
                        <valueType>scalar</valueType>
                        <parameterId>SSTG</parameterId>
                        <locationId>AUGM1ME</locationId>
                        <timeSeriesType>simulated historical</timeSeriesType>
                        <timeStep unit="hour" multiplier="1"/>
                        <readWriteMode>add originals</readWriteMode>
                    </timeSeriesSet>
                    <timeSeriesSet>

    <moduleInstanceId>HECRAS_KENNEBEC_UpdateStates</moduleInstanceId>
                        <valueType>scalar</valueType>
                        <parameterId>SQIN</parameterId>
                        <locationId>AUGM1ME</locationId>
                        <timeSeriesType>simulated historical</timeSeriesType>
                        <timeStep unit="hour" multiplier="1"/>
                        <readWriteMode>add originals</readWriteMode>
                    </timeSeriesSet>
                </timeSeriesSets>
            </importTimeSeriesActivity>
        </importActivities>
    </activities>
</generalAdapterRun>
```

Appendix 3: Parameter File


```
<parameters xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.wldelft.nl/fews/PI"
xsi:schemaLocation="http://www.wldelft.nl/fews/PI http://fews.wldelft.nl/schemas/version1.0/pi-
schemas/pi_modelparameters.xsd" version="1.5">

    <modifierType>HECRAS</modifierType>
    <group id="default" name="hec-ras run parameters">
        <parameter id="ComputationInterval">
            <description>Computation interval in minutes. Does not change interval of output data.</description>
            <intValue>60</intValue>
        </parameter>
    </group>
    <group id="Gate #1" name="hec-ras gate parameters">
        <!-- Gate name or locationId should equal to location of gate -->
        <locationId>CT River R1/220000</locationId>
        <!--

        Gate parameters depend on the mode selected in the RAS configuration files (gui),

        Posssible modes which can te set in the gui are:

            based on upstream WS (default)
            based on specified reference
            based on difference in stage
        -->
        <!-- parameters valid all modes -->
        <parameter id="RateOpen">
            <dblValue>0.05</dblValue>
        </parameter>
        <parameter id="RateClose">
            <dblValue>0.05</dblValue>
        </parameter>
        <parameter id="MaxOpen">
            <dblValue>20.0</dblValue>
        </parameter>
        <parameter id="MinOpen">
            <dblValue>0.0</dblValue>
        </parameter>
        <parameter id="InitOpen">
            <dblValue>3.0</dblValue>
        </parameter>
        <!-- parameters specific for "based on upstream WS" -->
        <parameter id="ZOpen">
            <description/>
            <dblValue>4.0</dblValue>
        </parameter>
```

```
<parameter id="ZClose">
    <description/>
    <dblValue>3.0</dblValue>
</parameter>
<!-- parameters specific for "based on specified reference" -->
<!--
<parameter id="ReferenceWSType">
    <description>Valid values: Reach, RiverStation, StorageArea</description>
    <stringValue>Reach</stringValue>
</parameter>

<parameter id="ReferenceWS">
    <description>Depending on the ReferenceWSType parameter</description>
    <stringValue>R1</stringValue>
</parameter>

<parameter id="referenceWSOpen">
    <description>Reference elevation at which gate begins to open</description>
    <dblValue>4.0</dblValue>
</parameter>

<parameter id="referenceWSClose">
    <description>Reference elevation at which gate begins to close</description>
    <dblValue>3.0</dblValue>
</parameter>
-->
<!-- parameters specific for "based on difference in stage" -->
<!--
<parameter id="stageDiffUSType">
    <description>Upstream Reach, RiverStation or StorageArea location for stage difference
computation</description>
    <stringValue>Reach</stringValue>
</parameter>

<parameter id="stageDiffUS">
    <description>Depends on the stageDiffUSType parameter</description>
    <stringValue>Reach</stringValue>
</parameter>

<parameter id="stageDiffDSType">
    <description>Downstream River, Reach, RiverStation or StorageArea location for stage difference
computation</description>
    <stringValue>Reach</stringValue>
</parameter>

<parameter id="stageDiffDS">
    <description>Depends on the stageDiffDSType parameter</description>
    <stringValue>R1</stringValue>
</parameter>

<parameter id="stageDiffOpen">
```

```
            <description>Stage difference at which gate begins to open</description>
            <dblValue>0.1</dblValue>
        </parameter>

        <parameter id="stageDiffClose">
            <description>Stage difference at which gate begins to close</description>
            <dblValue>0.1</dblValue>
        </parameter>
        -->
    </group>
    <group id="Gate #1" name="hec-ras gate parameters">
        <!-- Gate name or locationId should equal to location of gate -->
        <locationId>CT River R1/130000</locationId>
        <!--

            Gate parameters depend on the mode selected in the RAS configuration files (gui),

        Posssible modes which can te set in the gui are:

            based on upstream WS (default)
            based on specified reference
            based on difference in stage
        -->
        <!-- parameters valid all modes -->
        <parameter id="RateOpen">
            <dblValue>0.04</dblValue>
        </parameter>
        <parameter id="RateClose">
            <dblValue>0.04</dblValue>
        </parameter>
        <parameter id="MaxOpen">
            <dblValue>20.0</dblValue>
        </parameter>
        <parameter id="MinOpen">
            <dblValue>0.0</dblValue>
        </parameter>
        <parameter id="InitOpen">
            <dblValue>3.0</dblValue>
        </parameter>
        <!-- parameters specific for "based on upstream WS" -->
        <parameter id="ZOpen">
            <description/>
            <dblValue>4.0</dblValue>
        </parameter>
        <parameter id="ZClose">
            <description/>
            <dblValue>3.0</dblValue>
        </parameter>
    </group>
    <group id="Levee Breach" name="hec-ras levee breach parameters">
        <locationId>CT River R1/248658</locationId>
```

```
<parameter id="IsActive">
    <description>true when breach is activated, otherwise model skips it during computations</description>
    <boolValue>false</boolValue>
</parameter>
<parameter id="IsWSStart">
    <description>true if trigger for failure is WS elevation</description>
    <boolValue>true</boolValue>
</parameter>
<parameter id="ThresholdWS">
    <description>water surface elevation for breaching</description>
    <dblValue>3.4028E38</dblValue>
</parameter>
<parameter id="ThresholdDuration">
    <description>threshold time (hours) for breaching</description>
    <dblValue>3.4028E38</dblValue>
</parameter>
<parameter id="StartDate">
    <description>Start date for breaching (e.g. 01MAR2001)</description>
    <stringValue/>
</parameter>
<parameter id="StartTime">
    <description>Start time for breaching (e.g. 1630)</description>
    <stringValue/>
</parameter>
<parameter id="CenterStation">
    <description>Center of breach (XS station / location)</description>
    <dblValue>8800.0</dblValue>
</parameter>
<parameter id="BottomWidth">
    <description>Final bottom width</description>
    <dblValue>500.0</dblValue>
</parameter>
<parameter id="BottomElevation">
    <description>Final bottom elevation</description>
    <dblValue>-10.0</dblValue>
</parameter>
<parameter id="LeftSideSlope">
    <description>Left side slope</description>
    <dblValue>2.0</dblValue>
</parameter>
<parameter id="RightSideSlope">
    <description>Right side slope</description>
    <dblValue>2.0</dblValue>
</parameter>
<parameter id="BreachTime">
    <description>Full formation time (hours)</description>
    <dblValue>1.0</dblValue>
</parameter>
<parameter id="WeirCoef">
    <description>Breach weir coefficient</description>
    <dblValue>2.6</dblValue>
```

```
        </parameter>
        <!-- parameter below are used only when IsPipe = true -->
        <parameter id="IsPipe">
            <description>true if piping failure, false if overtopping</description>
            <boolValue>true</boolValue>
        </parameter>
        <parameter id="PipingCoefficient">
            <description>Piping coefficient (default is .8)</description>
            <dblValue>0.8</dblValue>
        </parameter>
        <parameter id="InitialPipingElevation">
            <description>Initial piping elevation</description>
            <dblValue>-0.5</dblValue>
        </parameter>
    </group>
</parameters>
```

Appendix 4: Preprocessing


```xml
<?xml version="1.0" encoding="UTF-8"?>
<transformationModule 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/transformationModule.xsd" version="1.0">
    <variable>
        <variableId>FLOW</variableId>
        <timeSeriesSet>
    <moduleInstanceId>CHANGET_AUGM1ME_SIDM1ME_UpdateStates</moduleInstanceId>
            <valueType>scalar</valueType>
            <parameterId>QINE</parameterId>
            <locationId>SIDM1ME</locationId>
            <timeSeriesType>simulated historical</timeSeriesType>
            <timeStep unit="hour" multiplier="1"/>
            <relativeViewPeriod unit="hour" start="-240" startOverrulable="true" end="0"/>
            <readWriteMode>read only</readWriteMode>
        </timeSeriesSet>
    </variable>
    <variable>
        <variableId>MIN_FLOW</variableId>
        <timeSeriesSet>
    <moduleInstanceId>HECRAS_KENNEBEC_Preprocessing_UpdateStates</moduleInstanceId>
            <valueType>scalar</valueType>
            <parameterId>QINE</parameterId>
            <locationId>SIDM1ME</locationId>
            <timeSeriesType>simulated historical</timeSeriesType>
            <timeStep unit="hour" multiplier="1"/>
            <relativeViewPeriod unit="hour" start="-240" startOverrulable="true" end="0"/>
            <readWriteMode>read only</readWriteMode>
        </timeSeriesSet>
    </variable>
    <variable>
        <variableId>STAGE</variableId>
        <timeSeriesSet>
    <moduleInstanceId>NOMSNG_AUGM1ME_CASM1_UpdateStates</moduleInstanceId>
            <valueType>scalar</valueType>
            <parameterId>STID</parameterId>
            <locationId>CASM1ME</locationId>
            <timeSeriesType>simulated historical</timeSeriesType>
            <timeStep unit="hour" multiplier="1"/>
            <relativeViewPeriod unit="hour" start="-240" startOverrulable="true" end="0"/>
            <readWriteMode>read only</readWriteMode>
        </timeSeriesSet>
    </variable>
    <variable>
        <variableId>MIN_STAGE</variableId>
```

```
        <timeSeriesSet>
    <moduleInstanceId>HECRAS_KENNEBEC_Preprocessing_UpdateStates</moduleInstanceId>
            <valueType>scalar</valueType>
            <parameterId>STID</parameterId>
            <locationId>CASM1ME</locationId>
            <timeSeriesType>simulated historical</timeSeriesType>
            <timeStep unit="hour" multiplier="1"/>
            <relativeViewPeriod unit="hour" start="-240" startOverrulable="true" end="0"/>
            <readWriteMode>read only</readWriteMode>
        </timeSeriesSet>
    </variable>
    <transformation id="set_miminum_flow">
        <rangeTransformation>
            <range>
                <limitVariableId>FLOW</limitVariableId>
                <lowerLimit>-100000</lowerLimit>
                <upperLimit>200</upperLimit>
            </range>
            <user>
                <simple>
                        <expression>200</expression>
                        <outputVariable>
                                <variableId>MIN_FLOW</variableId>
                        </outputVariable>
                </simple>
            </user>
        </rangeTransformation>
        <rangeTransformation>
            <range>
                <limitVariableId>FLOW</limitVariableId>
                <lowerLimit>200</lowerLimit>
                <upperLimit>1000000000</upperLimit>
            </range>
            <user>
                <simple>
                        <expression>FLOW</expression>
                        <outputVariable>
                                <variableId>MIN_FLOW</variableId>
                        </outputVariable>
                </simple>
            </user>
        </rangeTransformation>
    </transformation>
    <transformation id="InterpolateInputFlow">
        <interpolationSerial>
            <linear>
                <inputVariable>
                        <variableId>MIN_FLOW</variableId>
                </inputVariable>
                <maxGapLength>10</maxGapLength>
                <outputVariable>
```

```
                        <variableId>MIN_FLOW</variableId>
                    </outputVariable>
                </linear>
            </interpolationSerial>
        </transformation>
        <transformation id="InterpolateInputStage">
            <interpolationSerial>
                <linear>
                    <inputVariable>
                        <variableId>STAGE</variableId>
                    </inputVariable>
                    <maxGapLength>10</maxGapLength>
                    <outputVariable>
                        <variableId>MIN_STAGE</variableId>
                    </outputVariable>
                </linear>
            </interpolationSerial>
        </transformation>
</transformationModule>
```

Appendix 5: Filters


```
<filter id="hecras" name="HEC-RAS">
    <mapExtentId>NERFC_Connecticut</mapExtentId>
    <child foreignKey="hecras preprocessing"/>
    <child foreignKey="hecras output"/>
</filter>
<filter id="hecras preprocessing" name="hecras preprocessing">
    <mapExtentId>Connecticut River</mapExtentId>
    <timeSeriesSet>
        <moduleInstanceId>HECRAS_KENNEBEC_Preprocessing_Forecast</moduleInstanceId>
        <valueType>scalar</valueType>
        <parameterId>QINE</parameterId>
        <locationId>SIDM1ME</locationId>
        <timeSeriesType>simulated forecasting</timeSeriesType>
        <timeStep unit="hour" multiplier="1"/>
        <relativeViewPeriod unit="hour" start="-240" startOverrulable="true" end="240" endOverrulable="true"/>
        <readWriteMode>read only</readWriteMode>
    </timeSeriesSet>
    <timeSeriesSet>
        <moduleInstanceId>HECRAS_KENNEBEC_Preprocessing_Forecast</moduleInstanceId>
        <valueType>scalar</valueType>
        <parameterId>STID</parameterId>
        <locationId>CASM1ME</locationId>
        <timeSeriesType>simulated forecasting</timeSeriesType>
        <timeStep unit="hour" multiplier="1"/>
        <relativeViewPeriod unit="hour" start="-240" startOverrulable="true" end="240" endOverrulable="true"/>
        <readWriteMode>read only</readWriteMode>
    </timeSeriesSet>
</filter>
<filter id="hecras output" name="hecras output">
    <mapExtentId>Connecticut River</mapExtentId>
    <timeSeriesSet>
        <moduleInstanceId>HECRAS_KENNEBEC_UpdateStates</moduleInstanceId>
        <valueType>scalar</valueType>
        <parameterId>SSTG</parameterId>
        <locationId>AUGM1ME</locationId>
        <timeSeriesType>simulated historical</timeSeriesType>
        <timeStep unit="hour" multiplier="1"/>
        <relativeViewPeriod unit="hour" start="-240" startOverrulable="true" end="240" endOverrulable="true"/>
        <readWriteMode>add originals</readWriteMode>
    </timeSeriesSet>
    <timeSeriesSet>
        <moduleInstanceId>HECRAS_KENNEBEC_UpdateStates</moduleInstanceId>
        <valueType>scalar</valueType>
        <parameterId>SQIN</parameterId>
        <locationId>AUGM1ME</locationId>
        <timeSeriesType>simulated historical</timeSeriesType>
```

```
            <timeStep unit="hour" multiplier="1"/>
            <relativeViewPeriod unit="hour" start="-240" startOverrulable="true" end="240" endOverrulable="true"/>
            <readWriteMode>add originals</readWriteMode>
        </timeSeriesSet>
        <timeSeriesSet>
            <moduleInstanceId>HECRAS_KENNEBEC_Forecast</moduleInstanceId>
            <valueType>scalar</valueType>
            <parameterId>SSTG</parameterId>
            <locationId>AUGM1ME</locationId>
            <timeSeriesType>simulated forecasting</timeSeriesType>
            <timeStep unit="hour" multiplier="1"/>
            <relativeViewPeriod unit="hour" start="-240" startOverrulable="true" end="240" endOverrulable="true"/>
            <readWriteMode>add originals</readWriteMode>
        </timeSeriesSet>
        <timeSeriesSet>
            <moduleInstanceId>HECRAS_KENNEBEC_Forecast</moduleInstanceId>
            <valueType>scalar</valueType>
            <parameterId>SQIN</parameterId>
            <locationId>AUGM1ME</locationId>
            <timeSeriesType>simulated forecasting</timeSeriesType>
            <timeStep unit="hour" multiplier="1"/>
            <relativeViewPeriod unit="hour" start="-240" startOverrulable="true" end="240" endOverrulable="true"/>
            <readWriteMode>add originals</readWriteMode>
        </timeSeriesSet>
    </filter>
```