

CHPS OHD Core 4.5.a Release Notes

Release Date: 10/28/2016
Release Type: Scheduled
CHPS Build: 5.4.1
CHPS_OHD_CORE Build: 4.5.a
Build and Package Date: 10/24/2016
Tested against: FEWS: 2015.02 build 63713, patched from 59855

Introduction

These **Release Notes** provide the latest information about the OHD_CORE_CHPS software system. The following topics are discussed:

- Overview
- What's New
- Resources / Documentation
- Known Issues and Limitations
- Detailed Description of Software Changes and Enhancements

Overview

With this release the old Lag/K Parameter Names will be invalid and not work. If you have not already replaced/removed them, please do so at this time. The old Lag/K Parameter Names can be replaced/removed by using a Cleanup Script provided with this release. The script name is cleanUpParDefinitions and can be found in the release package under /ohd/scripts/.

A complete list may be found below in the *What's New* section and further details are available in the *Detailed Description* section.

What's New

Fixes

FogBugz ID	Redmine ID	Reported By	Title
1949	24993	NWC	ofsde: Remove mape calculation.
1973	25017	NCRFC	UZTWCADJ and LZTWCADJ double accounting
1976	25019	ABRFC	EnsPost produces sawtooth plots during baseflow situations.
1985	25028	NWC	Use modulation (canonical) events for precipitation by default.
2018	25059	NWC	Corruption of an Aggregation Unit (AU) when selecting multiple AUs
2019	25060	NWC	Failure to consistently remove invalid results from the Output window
2023	25064	NWC	HEFSEnsPost Season Selection Error

Enhancements

FogBugz ID	Redmine ID	Requested By	Title
1249	24395	ABRFC	GUI for CHANLOSS changes in Calibration mods
1651	24734	NWC	OHD modules cleanup script
1953	24997	NWRFC	Improve temporal aggregation of time-series with missing data.
1969	25013	CNRFC	Graphgen Fixing lower or upper axis bound only.
1988	25031	TVA	Update to OHDFEWSAdapter to handle 'NaN' values in Parameter files.
2005	25046	NWC	Lag/K old parameter names logic to be removed from code
2006	25047	NWC	Provide Calibration modules templates that reduce configuration size.
2040	25079	NWC	Create a Java version of the CHANLOSS model.

Resources / Documentation

Scripts

cleanUpParDefinitions

Documentation

The following pieces of documentation have been modified since the last release and can be found in the directory at the root of the package. All the CHPS documentation may be found online at <http://www.nws.noaa.gov/oh/hrl/general/indexdoc.htm#core>.

Version: OHD-CORE-CHPS-4.5.a

Release Date: 28 October 2016

- *Modified:*
Calibration Configuration Guide
Channel Loss
EVS Manual
Graphics Generator Reference Manual
Lag and K routing
ofsde CHPS doc
Sacramento Soil Moisture Accounting
- *New: None*
- *Removed: None*

Known Issues and Limitations

N/A

Detailed Description of Software Changes and Enhancements

Fixes:

FogBugz 1949 – *ofsde: Remove mape calculation*

Description

ofsde still has code to calculate Mean Areal Evapotranspiration (RC24) data. This calculation requires input from USERPARM OFS file. This file is read via a FORTRAN subroutine. The calculation of RC24 will be removed. This includes the removal of the FORTRAN subroutine from the compile/build.

Cause

Some old out-dated FORTRAN code has been left in ofsde.

Fix

Remove this old FORTAN code.

Notes

None.

Version: OHD-CORE-CHPS-4.5.a

Release Date: 28 October 2016

FogBugz 1973 – UZTWCADJ and LZTWCADJ double accounting

Description

When an UZTWCADJ or LZTWCADJ modifier is defined for 12Z on a particular day, the UpdateStates run incorporates the effect of the modifier into the warm state. However, when that particular day becomes the warm state start for a forecast run, the modifier is applied again effectively doubling, and it's effect and corrupting the warm states.

Cause

The UZTWCADJ and LZTWCADJ modifiers use the non-equidistant time series which applied twice if the modifier is defined for 12Z.

Fix

Corrected the logic in SACSMA for the UZTWCADJ and LZTWCADJ modifiers, and changed time step to an equidistant time series in ModuleConfigFiles.

Notes

The SACSMA Forecast and UpdateStates Module files (i.e., SACSMA_FGIX_Forecast.xml and SACSMA_FGIX_UpdateStates.xml) in ModuleConfigFiles directory need to be updated as following:

- 1 Delete <timeSeriesSet> blocks for parameterId UZTWCADJ and LZTWCADJ
- 2 Add lines below to <exportFile>mods_sacsma_eq.fi</exportFile> section after the <timeSeriesSets> tag.

```
<timeSeriesSet>
  <moduleInstanceId>ExportMODS</moduleInstanceId>
  <valueType>scalar</valueType>
  <parameterId>UZTWCADJ</parameterId>
  <locationId>XXXXX</locationId>
  <timeSeriesType>external historical</timeSeriesType>
  <timeStep multiplier="6" unit="hour"/>
  <relativeViewPeriod end="0" endOverrutable="true" unit="hour"/>
  <readWriteMode>read only</readWriteMode>
  <ensembleId>main</ensembleId>
</timeSeriesSet>
<timeSeriesSet>
  <moduleInstanceId>ExportMODS</moduleInstanceId>
  <valueType>scalar</valueType>
```

```
<parameterId>LZTWCADJ</parameterId>
<locationId>XXXXX</locationId>
<timeSeriesType>external historical</timeSeriesType>
<timeStep multiplier="6" unit="hour"/>
<relativeViewPeriod end="0" endOverrutable="true" unit="hour"/>
<readWriteMode>read only</readWriteMode>
<ensembleId>main</ensembleId>
</timeSeriesSet>
```

FogBugz 1976 – *EnsPost produces sawtooth plots during baseflow situations*

Description

Some of our locations are still producing consistent "sawtooth" plots from EnsPost especially during baseflow situations. FB 1516 fixed a majority of the old issues we had and this appears related but not the same case.

Cause

During parameter estimation, it is possible for the a-parameter, the coefficient applied to the observed value at the preceding time step, to be negative. If it becomes significantly negative, then an oscillation effect occurs within the ARX(1) model employed by EnsPost, resulting in a saw-tooth pattern.

Fix

EnsPostPE was restricted so that both coefficients of the linear model are restricted to the interval [0, 1].

Notes

To implement this fix, re-estimate the parameters for any segment for which the saw-tooth pattern to the output is observed. Alternatively, to identify the segments for which the saw-tooth pattern may occur, do the following (see the EnsPostPE Configuration Guide for location of `<enspostpe_run_area>`):

```
cd <enspostpe_run_area>
cd parameters
grep "_a_".param | grep "\-
```

Parameters should be re-estimated for any segments for which the last grep command returns a matching line.

FogBugz 1985 – *Use modulation (canonical) events for precipitation by default*

Description

Recent evidence from MARFC, NERFC, CNRFC, and WGRF indicates that modulation (canonical) events significantly improve the quality of the MEFP precipitation forecasts, particularly for higher precipitation amounts and larger accumulations (e.g. multi-day). Modulation events were eliminated from the MEFP default canonical events (in an earlier build), both for precipitation and temperature, as they were found to exacerbate problems with discontinuities at event boundaries (i.e. where the forecast quality shifted abruptly at event boundaries). However, these problems were largely confined to the temperature forecasts. Given the improvements in forecast quality, we therefore recommend to modify the MEFP defaults to include modulation events for precipitation, but not for temperature.

Cause

See Description.

Fix

Modulation canonical events have been added to the default canonical events delivered with the release of CHPS and MEFPE. They will be available with the release indicated as the milestone.

Notes

This change will automatically be picked up as part of any newly installed parameter estimation SA. **For existing parameter estimation SAs, [instructions are provided in the FogBugz post](#).**

For any MEFP location for which you want to include the modulation events, the MEFP parameters will need to be re-estimated and 'accepted' in order to put them in place for use by the MEFPEnsembleGeneratorModelAdapter.

FogBugz 2018 – *Corruption of an Aggregation Unit (AU) when selecting multiple AUs*

Description

When selecting multiple AUs in the first window of the Aggregation Stage, the parameters associated with the final AU in the selection were incorrectly assigned to those associated with the first AU in the selection. This led to an inconsistency between the saved state of the AUs and the displayed state, which propagated to (and corrupted) the local store of parameter values. Consequently, an error was thrown on attempting to save the project, as more than one AU had been (incorrectly) assigned with the same identifier (which is not allowed).

Cause

When creating a new AU or loading an existing AU, the parameters associated with that AU are added to a local store in the `evs.gui.windows.AggregationA.java`. This local store is coordinated with a change in the selected AU; that is in the table of AUs. Specifically, a change in the selected AU

Version: OHD-CORE-CHPS-4.5.a

Release Date: 28 October 2016

leads, first, to an update of the local store via the saveLocalData method, which records any changes to the parameters of the current selection. Second, the GUI is updated to show the parameters of the newly selected AU (from the local store) via the showLocalData method. Upon selecting multiple AUs, the index of the AU to save, versus the index of the AU to display, became inconsistent. This propagated to, and corrupted, the local store of parameter values for the previously selected AU.

Fix

Upon selecting a new AU in the table of AUs, the save and display operations were coordinated from within a listener attached to the table model. For improved clarity, the listener now calls a separate method, updateUnitSelection (in keeping with a similar approach in evs.gui.windows.VerificationA.java). The new method uses the anchor selection index to determine the currently selected AU and properly coordinates the save and display states within a multi-unit selection.

Notes

Tested the updates by adding and deleting AUs and by experimenting with the selection and reselection of AUs, including the selection of multiple AUs in sequence and separated by unselected AUs. In all cases, the GUI was updated properly and the local store of parameters remained consistent with the display state.

FogBugz 2019 – *Failure to consistently remove invalid results from the Output window*

Description

Verification results are displayed in the Output window of the EVS GUI. Upon changing one or more parameter values that invalidate these results, the results should be removed from the Output window. There should be no requirement to save any changes to the parameter values in order to trigger the removal of invalid results from the Output window. In practice, however, existing results were only updated (and, therefore, removed, when required) upon saving changes. Thus, unsaved changes were not immediately propagated to the Output window. This led to an inconsistency between the updated parameter values and the previously computed verification results, still available in the Output window.

Cause

Several dialogs within the EVS allow for changes to parameter values that may invalidate earlier verification results. These include the evs.gui.windows.MoreVerificationWindowDialog.java and the evs.gui.windows.MoreInputDialog.java. In both cases, the parameter values are updated upon closing the dialog. However, these changes were not propagated to the evs.gui.windows.OutputA.java.

Fix

Revised the saveData method of MoreVerificationWindowDialog.java and evs.gui.windows.OutputA.java to call the updateLocalData method of OutputA.java, in order to propagate any changes in parameter values to the Output window.

Notes

Tested the fix by, firstly, conducting verification (in order to generate verification results in the Output window) and, secondly, by changing various parameter values that should invalidate (remove) the previously computed results. In all cases, the earlier results were removed, once the changes were confirmed (i.e. once the dialogs were closed).

FogBugz 2023 – HEFSEnsPost Season Selection Error

Description

With the addition of stratified sampling to EnsPost, the fix for FB 1231 fails in some cases leading to incorrect parameters being applied for some parts of the forecast period.

Cause

A change made to allow for stratified random sampling, which required restructuring the error model software, broke the fix implemented for FB 1231 causing it to incorrectly identify the season.

Only under specific circumstances will this bug yield potential problems. First, the T0 must be within 15 days from the end of its month. Next, there must be at least one future month within the forecast period within the same calendar year as the T0 (i.e., no problems will occur for December T0s). Finally, there must be a change in parameter values due to a season boundary allowing for incorrect parameters to be used with the 15-day shifted time period. The incorrect parameters will then be used for those 15 days.

[See this post within the FogBugz for details.](#)

Fix

The season identification algorithm was corrected so that seasons are correctly identified according to the fix implemented for FB 1231. That fix is described in the link within the Cause section above.

Notes

None.

Enhancements:

FogBugz 1249 – *GUI for CHANLOSS Changes in Calibration Mods*

Description

Make a GUI to make changes to channel loss in the calibration mods.

Cause

N/A

Fix

The documentation for CHANLOSS was updated to describe how to configure a GUI for CHANLOSS changes.

Refer to section 5 of http://www.nws.noaa.gov/oh/hrl/general/chps/Models/Channel_Loss.pdf

Notes

The instructions assume the new Java version of CHANLOSS; therefore, the user should first update the CHANLOSS configuration files (refer to FogBugz 2040).

FogBugz 1953 – *Improve temporal aggregation of time-series with missing data*

Description

When aggregating a regular time-series that contains one or more missing values, the aggregation should be performed “on cycle”; that is, by including the missing values when determining the blocks to aggregate, rather than proceeding from the first available block without missing values. The consequences of aggregating “off cycle” are that some verification pairs will comprise “off cycle” forecast lead times. As the verification statistics are computed by pooling verification pairs with common forecast lead times, the statistics corresponding to “off-cycle” pairs generally comprise only small sample sizes. In practice, any missing values should be considered when determining the blocks for aggregation in a regular time-series, otherwise the aggregation will proceed on a variable cycle (i.e. depending on the distribution of missing values in each time-series).

Cause

By default, the aggregateTimeSeries method in evs.data.PairedDataUtilities.java conducts aggregation from the first non-missing value for which a valid aggregation block can be defined (a block whose values are all non-missing), regardless of whether that aggregation block is “on cycle”. In this context, an aggregation block comprising n verification pairs is “on cycle” if it begins xn timesteps after the first time for which aggregation is requested, where x is an integer greater than zero. By default, aggregation begins at the first non-missing time, but optionally (by user request), an absolute time, comprising either a forecast lead time or a valid time in UTC. Aggregation blocks that are “off cycle” (i.e. not “on cycle”) should be ignored.

Fix

Version: OHD-CORE-CHPS-4.5.a

Release Date: 28 October 2016

Updated the aggregateTimeSeries method in evs.data.PairedDataUtilities.java by abstracting out the logic for identifying aggregation blocks into a new method, getBlocksForAggregation. The new method identifies aggregation blocks that are "on cycle" and returns the first and last indices of these blocks for input to the updated aggregateTimeSeries method.

Notes

Tested the updates against synthetic time-series with different combinations of missing elements, as well as time-series with no missing elements.

FogBugz 1969 – *Graphgen Fixing lower or upper axis bound only*

Description

CNRFC would like to fix the lower y-axis bound to be zero, but leave the upper bound to be "auto" for some of their Graphics Generator plots. Right now, there doesn't appear to be a way to do this in Graphics Generator.

Cause

None.

Fix

These changes can be accomplished by allowing a setting of "auto" within the lower and/or upper bound values for numerical and date axis limits. In the XML, this may require assigning a value to indicate auto so that the XML element is still a numerical. In the GUI, the text fields may be changed to editable list that allow for specifying a number or selecting Auto. Specifying both as Auto would be equivalent to selecting full auto-computation of limits (i.e., the default for the axis limits).

Notes

None.

FogBugz 1988 – *Update to OHDFEWSAdapter to handle FEWS 'NaN' values in Parameters files.*

Description

TVA and other users of OHD-CORE are beginning to use moduleConfig and parameter templates to reduce their amount of configuration files. For parameter file templates, an issue arises when the number of valid parameter values needed/defined for a basin is less than the number of parameter value placeholders set in the template file. For this situation, the developers in Delft suggested to use 'NaN' as a "dummy value" to fill in the template for the extra parameters which are not needed. NaN values are already filtered out in the FEWS GUI.

Version: OHD-CORE-CHPS-4.5.a

Release Date: 28 October 2016

This enhancement request is to modify the OHD FEWS Adapter to handle the case when 'NaN' values are found in a parameter file and skip those values and continue on processing the parameter file.

Below is an extract example from a UNITHG model parameter file.

```
<parameter id="UHG_ORDINATES">
  <table>
    <columnTypes A="double"/>
    <row A="2793.1"/>
    <row A="5836.6"/>
    <row A="3264.3"/>
    <row A="1354.2"/>
    <row A="584.3"/>
    <row A="204.8"/>
    <row A="67.3"/>
    <row A="24.4"/>
    <row A="0.0"/>
    <row A="NaN"/>
    <row A="NaN"/>
  </table>
</parameter>
```

Internally CHPS will translate the original parameter information to be:

```
<parameter id="UHG_ORDINATES">
  <table>
    <columnTypes A="double"/>
    <row A="2793.1"/>
    <row A="5836.6"/>
    <row A="3264.3"/>
```

Version: OHD-CORE-CHPS-4.5.a

Release Date: 28 October 2016

```

    <row A="1354.2"/>
    <row A="584.3"/>
    <row A="204.8"/>
    <row A="67.3"/>
    <row A="24.4"/>
    <row A="0.0"/>
  </table>
</parameter>

```

This new parameter file could be read by the CHPS models without any error. While in the first case an error will be generated by the CHPS models.

This change was not currently handled by the OHDFewsadapter. In cases where the parameters files will not have 'NaN' values CHPS will continue to behave as is doing in previous versions.

Cause

None.

Fix

The template files (cvs format) will contain the parameters values including the parameters ordinates values for a specific location with NaN values set for some of the ordinates, for instance for location SMFV2 the template file will contain:

```

SMFV2,6,ENGLISH,6,131.3,0,9,2793.1,5836.6,3264.3,1354.2,584.3,204.8,67.3,24.4,0,0,0,N/A,N/A,N/A,N/A,N/A,
N/A,N/A,N/A,N/A,N/A,N/A,N/A,N/A,N/A,N/A,N/A,N/A,N/A,N/A,N/A,N/A,N/A,N/A,N/A,N/A,N/A,N/A,N/A,N/A,N/A,
N/A,N/A,N/A,N/A,N/A,N/A,N/A,N/A,N/A,N/A,N/A,N/A,N/A,N/A,N/A,N/A,N/A,N/A,N/A,N/A,N/A,N/A,N/A,N/A,
N/A,N/A,N/A,N/A,N/A,N/A,N/A,N/A,N/A,N/A,N/A,N/A,N/A,N/A,N/A,N/A,N/A,N/A,N/A,N/A,N/A,N/A,N/A,N/A,N/A,
A

```

These values got translated to a parameter XML file with the following values:

```

<parameter id="UHG_ORDINATES">
  <table>
    <columnTypes A="double"/>
    <row A="2793.1"/>
    <row A="5836.6"/>
    <row A="3264.3"/>
    <row A="1354.2"/>
    <row A="584.3"/>
  </table>
</parameter>

```



```
<row A="NaN"/>
<row A="NaN"/>
<row A="NaN"/>
<row A="NaN"/>
<row A="NaN"/>
</table>
</parameter>
```

....

The code changes to CHPS OHDFEWSAdapter remove the 'NaN' values from the parameters XML file that when executed it will internally translate the parameter file to:

...

```
<parameter id="UHG_ORDINATES">
```

```
<table>
  <columnTypes A="double"/>
  <row A="2793.1"/>
  <row A="5836.6"/>
  <row A="3264.3"/>
  <row A="1354.2"/>
  <row A="584.3"/>
  <row A="204.8"/>
  <row A="67.3"/>
  <row A="24.4"/>
  <row A="0.0"/>
  <row A="0.0"/>
  <row A="0.0"/>
</table>
</parameter>
```

....

Basically, the OHDFEWSAdapter enhancement will remove the 'NaN' values from the XML files before it is passed to the CHPS models to be executed.

Notes

None.

Version: OHD-CORE-CHPS-4.5.a

Release Date: 28 October 2016

FogBugz 1651 – *OHD modules cleanup script*

Description

During in-house alpha testing of OHD-CORE-4.5.a, OWP found the cleanUpParDefinitions script doesn't clean up the LAGK parameters properly for the CBRFC's SA.

Cause

N/A

Fix

The script is updated to handle additional subdirectories under ModuleParFiles and ColdStateFiles and corrected table columnType.

Please use the unclean configuration to run this script.

1. Place cleanUpParDefinitions script to Config/ directory
2. Run: ./cleanUpParDefinitions all

FogBugz 2005 – *Lag/K Old Parameter Names logic to be removed from code*

Description

OHD-CORE-4.4.a is the last Release that the old Lag/K Parameter Names will be valid and work. The source code will be updated in the OHD-CORE-4.5.a Release to remove the logic to allow for the old Lag/k Parameter Names.

Cause

N/A

Fix

The old Lag/K parameter names have been removed from the source code. The code will throw exception if the old parameter Id is found (i.e, ***“ERROR: From OHD-CORE 4.5.a, the old TSIDA parameter is no longer supported.”***).

Notes

If Lag/K model fail with error message above, then

The user should update the Lag/K parameters using the cleanup script (refer to FogBugz 1617), and

It may be necessary to “reinitialize” any “old” warm states created using the old Lag/K parameter names.

Version: OHD-CORE-CHPS-4.5.a

Release Date: 28 October 2016

FogBugz 2006 –Provide Calibration modules templates that reduce configuration size

Description

OWP will provide a set of Calibration modules templates that are an easy way to reduce the configuration size, simply workflows and ensure configuration consistency.

Cause

N/A

Fix

The Calibration configuration template files are included in OHD-CORE-4.5.a release package. The installation guide for Templates is documented in section 3 of Calibration Configuration Guide document.

Here is a list of CHPS Calibration module template files (ModuleConfigFiles/calibration/templates).

PEAKFLOW_Calibration_StatEquations.xml
PEAKFLOW_Convert_Peak.xml
PEAKFLOW_Selection_Calibration.xml
PEAKFLOW_Selection_Calibration_HOURS.xml
PEAKFLOW_Stats_Calibration.xml
Report_PEAKFLOW_Calibration.xml
Report_STATQME_Calibration.xml
STATQME_Calibration.xml
STATQME_Calibration_StatEquations.xml
STATQ_DAY.xml
STATQ_HOURS.xml
WATBAL_APICONT_Calibration.xml
WATBAL_SACSMA_Calibration.xml
WATBAL_SNOW17_Calibration.xml

Notes

If you have not yet set up CHPS calibration configurations (i.e. nwrfc_calb), then follows these instructions

http://www.nws.noaa.gov/oh/hrl/general/chps/Calibration/Calibration_Configuration_Guide.pdf

Otherwise, the instructions to update the existing calibration module are as follows:

Version: OHD-CORE-CHPS-4.5.a

Release Date: 28 October 2016

1. Directories of Note

- <release_dir>: the root directory of the untarred release package.
- (e.g., **OHD-CORE-CHPS-4.5.a/**)
- <template_dir>: the directory of template configuration files
- (e.g., <release_dir>/calibration/ConfigTEMPLATE)

- <Region_calb_dir>: the standalone directory used to develop your RFC calibrations (e.g., nwrfc_calb).
- <Config_dir>: <Region_calb_dir>/Config (e.g., nwrfc_calb/Config/)
- <ModuleConfigFiles_dir>: <Config_dir>/ModuleConfigFiles
- <RegionConfigFiles_dir>: <Config_dir>/RegionConfigFiles
- <ReportTemplateFiles_dir>: <Config_dir>/ReportTemplateFiles
- <SystemConfigFiles_dir>: <Config_dir>/SystemConfigFiles
- <WorkflowFiles_dir>: <Config_dir>/WorkflowFiles
- <Reports_dir>: <Region_calb_dir>/Reports

Configuration files in the <template_dir> directory contain further instructions about how to edit your existing files to make use of Calibration.

2. ModuleConfigFiles:

2a. Copy the set of provided Calibration modules xml templates to the calibration templates directory (e.g., ModuleConfigFiles/calibration/ templates).

```
$ cp -rf <template_dir>/ModuleConfigFiles/calibration/templates  
<ModuleConfigFiles_dir>/calibration/
```

2b. Check STATQ Calibration modules template (e.g., <ModuleConfigFiles_dir>/calibration/templates/STATQ_DAY.xml and STATQ_HOURS.xml) for correct relativeViewPeriod end time references

If your site runs calibration by setting T0 to a date/time with "00Z" then you should replace the provided STATQ template module configuration files like the following:

Change:

```
<relativeViewPeriod unit="hour" end="-24"/>
```

To:

```
<relativeViewPeriod unit="hour" end="0"/>
```

Note: These steps are ONLY needed for first time update configuration for FBz 2006.

3. WorkflowFiles:

Version: OHD-CORE-CHPS-4.5.a

Release Date: 28 October 2016

3a. Pick an existing workflow file to use the new template feature.

3b. Copy the provided template calibration workflow xml file to the existing workflow file was chosen.

```
$ cp -f  
<template_dir>/WorkflowFiles/calibration/XXXXX_Stats_Calibration.xml  
<WorkflowFiles_dir>/calibration/
```

There are several sample workflow templates; Use the one that most closely matches the analysis you wants to do.

See template files at <template_dir>/WorkflowFiles/calibration/

CREC1_Stats_Calibration.xml -

- run PEAKFLOW_Selection_Calibration_HOURS to compute observed peak using FEWS transformation, and
- STATQ_HOURS module for 1-hour time step

HETC1_Stats_Calibration.xml -

- run PEAKFLOW_Convert_Peak using USGS peak, and
- STATQ_DAY module for 24-hour time interval

SLYPLAWK_Stats_Calibration.xml -

- run PEAKFLOW_Selection_Calibration,
- WATBAL_APICONT_Calibration, and
- STATQ_DAY module for 24-hour time interval

3c. Edit Stats_Calibration workflow (from step 3b) and correct the values for all properties (e.g. GAGE, SEGMENT, TS (time step) SQIN_MODULE_ID, QME_MODULE_ID, OBSPEAK_MODULE_ID, DAILYORHOURLY, SQIN_LOC...).

For example: The values for all properties TS, GAGE, SEGMENT, SQIN_LOC, QME_MODULE_ID, and SQIN_MODULE_ID are from the STATQME_Calibration module shown in the snippet below.

```
<activity>  
  <properties>  
    <string key="TS" value="6"/>  
    <string key="GAGE" value="HETC1"/>  
    <string key="SEGMENT" value="HETC1H"/>
```

Version: OHD-CORE-CHPS-4.5.a

Release Date: 28 October 2016

```

    <string key="SQIN_LOC" value="HETC1"/>
    <string key="QME_MODULE_ID" value="ImportDataCard"/>
    <string key="SQIN_MODULE_ID"
value="LAGK_HETC1H_LAG_HET_UpdateStates"/>
  </properties>
  <runIndependent>false</runIndependent>
  <moduleInstanceId>STATQME_HETC1H_Calibration</moduleInstanceId>
  <moduleConfigFileName>STATQME_Calibration</moduleConfigFileName>
</activity>

```

3c. Check the Stats_Calibration workflow (from step 3b) for

- A correct value for the "SQIN_MODULE_ID" property; Set to the moduleInstanceId for the module where the SQIN data is coming from.

For example, SQIN in HETC1 is from the **LAGK** module shown in the snippet below. The module could **be ADDSUB, UNITHG**, etc. Please check and change the module accordingly. Check that the time step is correct for your configuration as well.

```

<timeSeriesSet>
  <moduleInstanceId>LAGK_HETC1H_LAG_HET_UpdateStates</moduleInstanceId>
  ...
  <parameterId>SQIN</parameterId>
  ...
  <timeStep unit="hour" multiplier="6"/>
</timeSeriesSet>

```

- A correct value for the "OBSPEAK_MODULE_ID" property; Set to the "PEAKFLOW_XXXXX_Convert_Peak" module if the PEAKFLOW_Convert_Peak module is used.

See template file at the location below.

```
<template_dir>/WorkflowFiles/calibration/HETC1_Stats_Calibration.xml
```

- A correct value for the "RSELCATCHMENT" property value
- A correct setting for the "DAILYORHOURLY" property

Note: *In order to do an hourly Peakflow analysis (vs a daily Peakflow analysis) a property key "DAILYORHOURLY" should be set to "HOURLY" or "DAILY"*

See template file at the location below.

```
<template_dir>/WorkflowFiles/calibration/HETC1_Stats_Calibration.xml
(DAILYORHOURLY set to "DAILY")
```

Or

```
<template_dir>/WorkflowFiles/calibration/CREC1_Stats_Calibration.xml  
(DAILYORHOURLY set to "HOURLY")
```

4. RegionConfigFiles:

- **Update the ModuleInstanceDescriptors, and ModuleInstanceSets to reflect the changes.**

See template file at the location below.

```
<template_dir>/RegionConfigFiles/ModuleInstanceDescriptors.xmltempla  
te and ModuleInstanceSets.xmltemplate
```

5. Launch the Stand alone (SA) and correct any configuration errors.

6. Repeat step 3, 4 and 5 for another segment.

FogBugz 2040 – *Create a Java version of the CHANLOSS model*

Description

To be able to satisfy the Enhancement Request to create a GUI for making changes to Channel Loss in the Calibration mods (FBz [1949](#)), we first need to create a Java version of the CHANLOSS model.

Cause

None.

Fix

The CHANLOSS hydrologic model was ported from Fortran to Java. The advantages of this version over the Fortran version are:

1. Improved performance (i.e. runtime) of running RFC workflows
2. Allows for future calibration of CHANLOSS using FEWS tools
3. Makes the Java model execution in CHPS platform independent
4. The Java code is easier to maintain.

Notes

- Configuration changes are needed to activate the new Java version of CHANLOSS. If the configurations are not changed, the existing Fortran model will continue to run.
- OWP is available to make the necessary configuration changes. We can provide a working stand-alone (SA) with the changes. It will be up to the RFCs to validate the runs and upload the changes to the "live" system

Version: OHD-CORE-CHPS-4.5.a

Release Date: 28 October 2016

- The Java version of the Chanloss model now uses the individual xml elements in the ModuleParameter files (i.e. **not the OPERATION_CONTENTS block**). This block can be removed later when the java version is validated.
- Testing showed the results of the new Java models differ in the tenths or hundredths place from the previous Fortran version.
- Testing showed that in some cases, the initial warm state runs of the Java version of CHANLOSS may fail. It may be necessary to “reinitialize” any “old” warm states created using the Fortran version of CHANLOSS. To do this, (assuming the cold states are reasonable), you can run an update state run initialized from a cold state. This run must start far enough back in the past to overwrite all the existing warm states.
- We came up with a solution where you can run the FORTRAN version of CHANLOSS executable on the converted parameters files (which contain the parameters for FORTRAN and Java format) and then after ALL of the warm States are updated through the FORTRAN version of CHANLOSS, you can just switch over to running the java version of CHANLOSS. The Java version of CHANLOSS can be run from then on.
- If you decide to follow the approach to keep the FORTRAN version of CHANLOSS a for a while please update the Configuration files provide by OWP for you RFC and after a while you will need to change the Configuration files to use the Java version of CHANLOSS (Step 6 of following instructions).
- If you want to move to 4.5.a. without keeping the FORTRAN version here is the practical way to resolve updating the CHANLOSS warm states.
 1. Update your Config directory with the update copy with migrated parameters files.
 2. Start FEWS
 3. Set the Current System time as T0 - 1.
 4. Open the "Tools-->Manual Forecast" Panel.
 5. Select the XXRFC UpdateStates (Carry Over) workflow.
Where XX is the two letters initial of a RFC, select "Cold state" radio button under "Select initial state" check box.
Click the "Run" button.
 6. Convert Config/ModuleConfigFiles to use CHANLOSS Java Models. using the command (be sure you run this command when you are in the Config/ModuleConfigFiles directory):


```
sed -i -
's|chanloss.ChanlossModelDriver|chanloss.ChanlossJavaModelDriver|g'
*/CHANLOSS*.xml
```
 7. Reload config.
 8. Reset the Current System time to the original T0 date.
 9. Done