

# HEFS-0.3.2 Release Notes

Release Date: 4/9/2013

Release Type: Development Build

HEFS Build: 0.3.2

Build and Package Date: 4/9/2013

Tested against FEWS Binary: 2012.02 build 39282 (patched from 38987)

## Introduction

This document contains release notes for HEFS-0.3.2.

- This HEFS release assumes the latest version of FEWS, made available with CHPS-3.0.1 in March 2013, (2012.02 build 39282 (patched from 38987) is installed on the machines used to execute HEFS.
- This release of HEFS is assumed to be “starting from scratch” in regards to FEWS configuration files. Previous versions of HEFS configurations should be saved off for reference, but none of the old configurations are assumed to be part of the initial configuration you will be adding HEFS-0.3.2 configurations to. It is recommended to start from a set of “operational” configurations that have been successfully tested against CHPS-3.0.1
- Refer to Appendix B: Repackaging existing EnsPost Parameters in 'EnsPostPEConfigurationGuide.docx' for information on how to save your EnsPost parameters from Development Release 2. You will be able to convert these parameters and save time with the Development Release 3 installation.
- The current approach for implementing HEFS uses 5 configuration guides. All but two of those (MEFPPE and ENSPOSTPE) include synchronization steps to upload configurations verified on an SA to the live system.

**Note: If you installed HEFS-0.3.1 (e.g. CNRFC), you will only need to go through 4 configuration guides. You will skip the MEFP Configuration Guide – Data Ingest Components.**

An alternative approach is to skip the intermediate synchronization steps and verify a complete end to end HEFS forecast on a stand-alone (SA) before synchronizing. This will be especially useful for minimizing the amount of downtime at MARFC and NERFC.

- HEFS-0.3.2 is a significant overhaul of the previous HEFS development releases. Old configurations are deleted by ensuring you start HEFS-0.3.2 from a new snapshot of your operational configurations (i.e. they should not have reference to HEFS configurations). **Remember to save off a version of the HEFS development release #2 (HEFS-0.2.2) configuration files for reference, some files/sections will be reused completely.**

Other cleanup includes:

- a) Removing the “opercfs”, “opergfs”, and “control\_template directories from \$MEFP\_ROOT\_DIR

- b) Deleting old scheduled workflows (ImportAndConvertHEFSGrids, HEFS\_EndToEnd\_Forecast)
  - c) Modifying an purge scripts used to maintain 30 days of ascii grid data in “opercfs” and 5 days of data in “opergfs”.
- After HEFS-0.3.2 has been installed and configured, please use the six tests in the Test Manual to verify that MEFP Data Ingest, MEFP PE, EnsPost PE, MEFP Forecast EnsPost, and GraphGen software is working.
  - The purge script delivered in Phase 1 with the data ingest components (.../dataIngest/ftpDir/scripts/purge\_cfsv2\_gfs\_gefs.sh) has been modified to purge old CFSv2 location-specific time series files created during CFSv2 data ingest.
- The script allows for a second argument that, if specified, indicates the location of the <mefp\_root\_dir> where the files will be purged. In that case, files with a last modified time more than 30 days old are removed. If not specified, old CFSv2 location-specific files will not be removed. If a cron entry for executing the purge script has already been created, then it must be changed to include the <mefp\_root\_dir> argument; see Section 2.7.3 of the *MEFP Configuration Guide: Data Ingest Components*.
- If needed, please refer to the updated “Graphics Generator Tips and Troubleshooting” document attached to FogBugz #1078. Three new sections were added to the original version of this document delivered with CHPS-3.0.1.

## Fixes

FogBugz ID	Reported By	Title
907	Mark Fresch	MEFP GFS+CFSv2 bug overwriting data
923	ABRFC	adding a new point to MEFP
945	NERFC	EnsPost Ex3a 4.7: Problem creating monthly plots to set seasons
948	CNRFC	MEFP workflow error in SA
963	ABRFC	MEFPPE Cannot estimate at LEPO2
970	CNRFC	ENSPOST - Dfferent location ID recognition
982	ABRFC	extract RFC QPF data in MEFPPE
986	NERFC	Trouble importing with workflow "ImportHEFSHistoricalData"
1007	ABRFC	GFS adjusted temps appear to be too low
N/A	OHD	EVS: Box plots failed to note the real units of measurement
N/A	CBRFC	EVS: Incorrect copying of thresholds for verification metrics
N/A	OHD	EVS: EVS html help documents not displayed correctly
N/A	OHD	EVS: Inability to condition dates on “week 53” during leap years
N/A	OHD	EVS: Error on sub-selecting lead-times for verification results
N/A	OHD	EVS: Timezone offsets other than 0 UTC in PI-XML files

## Enhancements

FogBugz ID	Reported By	Title
N/A	OHD	EVS: Enhance options for computing climatological probabilities
N/A	OHD	EVS: Inability to use relative paths for input and output data sources
N/A	CBRFC	EVS: Ambiguous labeling of "Start" button in Output window
N/A	OHD	EVS: Too many "Save" buttons in EVS GUI, causing confusion
N/A	OHD	EVS: Exit options could lead to inadvertent failure to save project
N/A	OHD	EVS: Inadequate warning messages for custom ASCII date formats
N/A	OHD	EVS: Allow custom measurement units for temporal scale
N/A	OHD	EVS: Redundant computation of duplicate thresholds
N/A	OHD	EVS: Handling of large datasets at aggregated resolutions
N/A	OHD	EVS: Use of NWS binary ("CS") files for verifying observations
N/A	OHD	EVS: Inefficient sorting of very large datasets
N/A	OHD	EVS: Use of not-a-number (NaN) as "no data" values in the EVS

## Documentation

The following pieces of documentation have been added or modified since the last release and can be found in the 'documentation' directory at the root of the package.

- *Added:* install\_notes/ HEFS-0.3.2\_EnsPostPEConfigurationGuide.pdf
- *Added:* install\_notes/ HEFS-0.3.2\_EnsPostConfigurationGuide.pdf
- *Modified:* install\_notes/ HEFS-0.3.2\_MEFPConfigurationGuide\_DataIngest.pdf
- *Added:* install\_notes/ HEFS-0.3.2\_MEFPConfigurationGuide\_Forecast.pdf
- *Added:* install\_notes/ HEFS-0.3.2\_MEFPPEConfigurationGuide.pdf
- *Added:* install\_notes/ HEFS-0.3.2\_HEFSGraphicsGeneratorProductsInstallationGuide.pdf
- *Added:* install\_notes/ HEFS-0.3.2\_HindcastingGuide.pdf
- *Added:* install\_notes/HEFS-0.3.2\_install\_notes.pdf
- *Modified:* user\_manuels/ HEFS-0.3.2\_MEFP\_Users\_Manual.pdf
- *Modified:* user\_manuels/ HEFS-0.3.2\_EnsPost\_Users\_Manual.pdf
- *Modified:* user\_manuels/ HEFS-0.3.2\_EVS\_MANUAL.pdf
- *Added:* user\_manuels/ HEFS-0.3.2\_HEFSOverviewAndGettingStarted.pdf
- *Added:* release\_notes/HEFS-0.3.2\_release\_notes.pdf
- *Added:* testing/HEFS-0.3.2/HEFS-0.3.2\_Feedback\_Form.pdf
- *Added:* testing/HEFS-0.3.2/HEFS-0.3.2\_Test\_Manual.pdf

## Notes

*Fix: Fogbugz 907 – MEFP GFS + CFSv2 bug overwriting data*

---

### **Description**

If CFS is used as a forcing in the FORTRAN version of MEFP (HEFS Dev. Release 2 or earlier, the CFS tmax forecast array actually contains the CFS tmin data. If GFS is used in conjunction, part or all of the CFS tmax forecast array is overwritten by the CFS tmax data. In addition to this, some of the GFS tmax data was corrupted because of a wrong FGS tmax array index.

**Cause**

In the FORTRAN version of MEFP (HEFS Dev. Release 2 or earlier), CFS tmax data was accidentally assigned to the GFS tmax array and the GFS tmax array index was wrong.

**Fix**

The code was changed to store both CFS and GFS data properly, also corrected the index used for GFS tmax array.

**Notes**

This case was fixed in HEFS-0.2.2 for FORTRAN and it has recently been ported to Java. OHD recommends the ensembles be reviewed to ensure such behavior has not made it into the Java version of the code.

---

***Fix: Fogbugz 923 – adding a new point to MEFP***

---

**Description**

MEFP was not generating ensemble time series of a new location by just adding the location id to the location set and creating a new module data set file that describes its forecast group and tributary name.

**Cause**

If parameter files are stored as module data set files, each module data set file needs to be indicated in a separate module configuration file.

**Fix**

The code was changed to read parameter files under MEFP\_ROOT\_DIR not from ModuleDataSetFiles.

**Notes**

Due to the data set file size limitation, the parameter files are not stored as module data sets. Rather, they are stored under the MEFP\_ROOT\_DIR. Each file is a gzipped-tar file (.tgz) containing mostly XML files specifying the parameters, making them human readable. The name of the file will be determined by the location id of the historical MAP time series used in parameter estimation and the data type of either "precipitation" or "temperature".

There is no longer any need for exportDataSetActivity in the MEFP modules. To find the parameter file that will be used by MEFP to generate an ensemble, the time series provided to the MEFP (QPF, GFS, etc.) will need to have the same locationId as the historical MAP time series. Id-mapping may be required.

---

***Fix: Fogbugz 945 – EnsPost Ex3a 4.7: Problem creating monthly plots to set seasons***

---

**Description**

The SQME and ESQME data in EnsPost PE after running Import and Flow Stats workflows is missing. As a result, monthly plots in the establishment of seasonality to flow corrections cannot be generated.

**Cause**

WorkflowDescriptors.xml entry for the HEFSFlowStats workflow did not have the timezone specified.

**Fix**

Added the following line to the HEFSFlowStats entry in WorkflowDescriptors.xml:

```
<timeZone>GMT-6</timeZone>
```

**Notes**

The Flow Stats workflow expects the datacard data to be in GMT-6.

---

***Fix: Fogbugz 948 – MEFP workflow error in SA***

---

**Description**

Problems were encountered with the wrong system time being determined by CHPS when executing MEFP and the interaction of the workflow forecast length with the output of MEFP was unclear.

**Cause**

The CHPS configuration was allowing for the input series to MEFP to have its length dictated by the forecast length of the workflow. Also, the first value exported was one step after T0, which confuses CHPS/FEWS into thinking that the system time of the module run for MEFP was one step after the true system time.

**Fix**

Configuration files have been constructed so that the end of all exported time series to MEFP cannot be overruled; i.e., it is fixed regardless of the workflow's forecast time. Also, all exported time series start from 0 (T0) instead of one step after. This should allow CHPS to identify the correct system time and always produce ensembles of the same length regardless of the workflow forecast length.

---

***Fix: Fogbugz 963 – MEFPE Cannot estimate at LEPO2***

---

**Description**

Parameter estimation was failing for the CFSv2 forecast source.

**Cause**

Java and the wrapped FORTRAN code were executing an algorithm to find the nearest CFSv2 point given the lat/lon for a catchment and coming up with different results due to some rounding issues.

**Fix**

The FORTRAN code is rewritten in Java, so that both use the same algorithm to find the nearest CFSv2 point.

**Notes**

ABRFC was told to round the location lat/lon for LEPO2 as a work around. Before testing this fix, be sure to recover the correct lat/lon for LEPO2: Lon: -95.585, Lat:36.8513888889 (based on possibly old configurations).

## *Fix: Fogbugz 970 – ENSPOST - Different location ID recognition*

---

### **Description**

Parameter files are not being properly identified by EnsPost due to an incorrectly constructed file name.

### **Cause**

EnsPost and EnsPostPE both construct parameter file names based on input time series (ensemble to post-process in EnsPost and provided historical simulated in EnsPostPE). If the locationIds or parameterIds do not match for the time series, then EnsPost may end up looking for a parameter file with a name that does not match the estimated parameters. A work-around is to use id-mapping in EnsPost.

### **Fix**

Multiple mechanisms are provided in EnsPost to specify the name of the parameter file:

- Default name is still constructed using locationId and parameterId of ensemble to post-process. Id-mapping can be used if this does not match those used in parameter estimation.
- The locationId and parameterId to use in constructing the parameter file name can be specified as run Info-file properties.
- The full parameter file name can be specified as a run-file property. This solution requires the smallest amount of configuration changes (one property must be specified; no id-mapping needed).

## *Fix: Fogbugz 982 – extract RFC QPF data in MEFPPE*

---

### **Description**

MEFPPE cannot extract precipitation forecast-observed pairs from the archive database used in constructing RFC forecast sourced archived forecasts.

### **Cause**

Various small bugs, including missing data in the vfpairs table not being accounted for by MEFPPE, and not properly handling basistimes stored with the pairs not matching exactly the corresponding forecast time (system time, T0) of the forecast data.

### **Fix**

The pairing algorithm has been rewritten to fix issues with handling missing data. Additionally, to identify the forecast time of a pair given the pair's basistime, a window is used; see the details of the FogBugz entry for the exact algorithm.

### **Notes**

Tests passed using both NERFC and ABRFC provided vfpairs data.

---

### *Fix: Fogbugz 986 – Trouble importing with workflow "ImportHEFSHistoricalData"*

---

#### **Description**

Problems were encountered in MEFPE when importing the datacard data. In addition, the "Found?" column of the **Select Time Series Dialogue** box that opens after pressing the export button is misleading.

#### **Cause**

The datacard files were being placed in the wrong directory for importing. The column header in the **Select Time Series Dialog** was poorly worded, causing users to think it indicates if the data was found in the CHPS database, when it was supposed to indicate if the time series was found in the export target directory.

#### **Fix**

Changed the column header to "Done?", reflecting if the export has already been done for that catchment/location.

#### **Notes**

A small enhancement request also came out of this bug. The small enhancement has *not* been completed and will be prioritized with everything else when determining upcoming development tasks.

---

### *Fix: Fogbugz 1007 – GFS adjusted temps appear to be too low*

---

#### **Description**

When using GFS as the only forecast source, the MEFPE generated temperature ensembles appeared unreasonably low (as low as -70C). While investigating the problem, a plotting error was discovered in which the 24-hour min/max plot was mislabeled, appearing off by 12-hours.

#### **Cause**

First, the datacard data used to estimate the parameters was off slightly, leading to bad parameters and, consequently, bad generated ensembles. Second, even after fixing, the value was still unreasonably low (-30C... not as low as before). This may be due to the diurnal equations being incorrect for ABRFC.

#### **Fix**

For the plotting error, all products were changed to have a 12Z axis starting point. Also, all plots were changed to GMT for simplicity. For the potential diurnal equation problem, this release uses CHPS transformations to perform all diurnal computations, meaning that ABRFC can modify the diurnal pattern as desired.

#### **Notes**

This problem requires further investigation after installing the new release. Does the problem persist? Could this be bad data, stated as a possibility in the bug? In the end, this may require a fundamental change in how the 24-hour TMIN/TMAX values are computed from 6-hour MAT.

# Notes for EVS Bug Fixes and Enhancements

## *Fix: Incorrect interpretation of dates in ASCII files*

---

### **Description**

When reading ASCII files containing forecasts or observations, the dates were interpreted with leniency. For example, 200005 was leniently interpreted when the required format was yyyyMMddHH, despite the missing ddHH. Instead of failing with an error message, the dates were read and wrongly interpreted without warning.

### **Cause**

The Java Calendar API has two modes for interpreting the calendar fields, lenient and non-lenient. The option to interpret dates leniently was wrongly being adopted by the `evs.utilities.ThreadSafeSimpleDateFormat`.

### **Fix**

The class that reads and interprets dates was updated to use a non-lenient interpretation of date strings.

## *Fix: Box plots failed to note the real units of measurement*

---

### **Description**

The box plots failed to note the real units of measurement of the forecast errors on the range axis (y-axis).

### **Cause**

Failure to append the units of measurements to the range axis in the box plots.

### **Fix**

Append the measurement units to the range axis in the box plots and require all box plot classes to implement the `evs.products.plots.defaultsRealValuedPlot`, which requires the plotting of real measurement units on the range axis, i.e. implementing `setRealUnits`.

## *Fix: Incorrect copying of thresholds for verification metrics*

---

### **Description**

In the second window of the Verification stage, clicking “Do all” to apply the thresholds for the currently selected verification metrics across all other metrics required the thresholds in the table to be finalized (i.e. the table row to have stopped editing) otherwise erroneous thresholds were reproduced across all metrics.

### **Cause**

Failure to finalize editing on the table of thresholds prior to using information contained in the table.

**Fix**

Finalize editing on the table of thresholds prior to using that information.

***Fix: EVS html help documents not displayed correctly***

---

---

**Description**

The EVS help documents for the verification metrics (accessed from the second window of the Verification stage) could not be opened in an external browser.

**Cause**

Inability of the method that locates the help documents for display to access those documents within the EVS.jar distribution. Use of a path within a jar file only worked if the files were provided in separate directories adjacent to the jar file with the same directory structure.

**Fix**

The solution involved decompressing the html documentation on-the-fly from the EVS.jar and then pointing the browser to the decompressed files for display. All help documentation can now be viewed on-the-fly in an external web browser.

***Fix: Inability to condition dates on "week 53" during leap years***

---

---

**Description**

When defining date condition from which to sub-select paired data for verification, week 53 was not supported as a conditioning option in the GUI (this can occur in leap years).

**Cause**

Date conditions are defined in the GUI, `2c. Set time parameters > More > Date condition`. The option to condition on calendar weeks of the year (whether by issue time or valid time) did not contain week 53.

**Fix**

Added week 53 to the list of available calendar weeks on which to condition.

***Fix: Error on sub-selecting lead-times for verification results***

---

---

**Description**

An exception was thrown on highlighting a subset of lead times for display in the Output window and then using the shortcut option from the context menu (accessed by a right click on the mouse), `Select highlighted times`.

**Cause**

The incorrect casting of lead times in the table of lead times from Double to Integer values, resulting in a `ClassCastException` at runtime.

**Fix**

Corrected the casting of values in the table of lead times to Double values.

***Fix: Timezone offsets other than 0 UTC in PI-XML files***

---

---

**Description**

Timezone offsets other than UTC (zero offset) in PI-XML files were ignored and an offset of zero UTC assumed. Also, decimal offsets of whole hours (e.g. 5.0) were interpreted as 0 UTC (a separate issue).

**Cause**

The handler for the PI-XML was not reading and interpreting an offset from UTC other than 0 correctly.

**Fix**

The PI-XML handler was fixed to allow for offsets from UTC other than zero hours and for whole hour decimals (although integer hours would normally be used in practice).

***Enhancement: Enhance options for computing climatological probabilities***

---

---

**Description**

When deriving climatological probabilities from a specified source of observed data, the climatological probabilities were derived from the paired observations before applying any date or value conditions. There was no option to derive the climatological probabilities from all available observations or to control how date and value conditions are applied to the observations.

**Cause**

Lack of functionality to control the observations used to derive climatological probabilities, specifically whether to use the paired observations or all observations and whether to apply date and value conditions to the observations.

**Fix**

Added functionality to control whether climatological probabilities are derived from the paired observations or all observations and whether date or value conditions are applied to the observations before deriving those probabilities.

**Notes**

To control whether all observations are used: using the GUI, 2b. Identify input data sources > More > Other options > Use all observations (not just paired) to determine climate thresholds; and from the EVS project file using the XML tag <use\_all\_observations\_for\_climatology>. To control whether date conditions are applied: using the GUI, 2b. Identify input data sources > More > Other options > Apply date conditions when determining climate thresholds; and from the EVS project file using the XML tag <apply\_date\_cond\_to\_climatology>. To control whether value conditions are applied: using the GUI, 2b. Identify input data

sources > More > Other options > Apply value conditions when determining climate thresholds; and from the EVS project file using the XML tag `<apply_value_cond_to_climatology>`.

---

### *Enhancement: Inability to use relative paths for input and output data sources*

---

#### **Description**

Relative paths are convenient when moving EVS project files and associated data directories but maintaining the relative path structure. Previously, only absolute paths were allowed. Paths relative to the EVS.jar were required for the input and output data directories.

#### **Cause**

Inability to specify relative paths for the input and output data directories.

#### **Fix**

Allowed relative paths for the input data directories and files, comprising the forecasts and observations, and for the output data. The paths are determined relative to the EVS.jar.

---

### *Enhancement: Ambiguous labeling of "Start" button in Output window*

---

#### **Description**

Labeling of the "Start" button in the Output window for returning to the first window was often confused with the "Run" button for generating graphical and numerical products.

#### **Cause**

Ambiguous labeling of the "Start" button in the Output window.

#### **Fix**

The "Start" button was re-labeled in the Output window to read "First."

---

### *Enhancement: Too many "Save" buttons in EVS GUI, causing confusion*

---

#### **Description**

Too many "Save" buttons in the GUI leading to confusion about the precise functionality of each "Save" and when they should be used.

#### **Cause**

Too many "Save" buttons.

#### **Fix**

Removed the individual "Save" buttons from each GUI window, as the "Save" option is always accessible from the task bar. This avoids confusion about when to save a project.

### *Enhancement: Exit options could lead to inadvertent failure to save project*

---

#### **Description**

Previously, on exiting the EVS GUI, there were options to exit (without save) or to cancel and return. This could lead to inadvertently exiting without saving the current project.

#### **Cause**

Unclear exit sequence, without an immediate option to save the current project, instead requiring the exit to be cancelled before implementing a save (i.e. the emphasis was on instigating a save before an exit).

#### **Fix**

An additional option was added to the exit sequence, namely saving the existing project before exiting. The three options now comprise exiting without saving, saving before exiting or cancelling.

### *Enhancement: Inadequate warning messages for custom ASCII date formats*

---

#### **Description**

Inadequate warning messages provided when specifying date formats for reading ASCII files. The date elements are case sensitive and failure to use case-sensitive elements could lead to unexpected behaviors. For example, 'm' is interpreted as minute whereas 'M' is interpreted as month.

#### **Cause**

Inadequate warning messages for user-defined date format strings. User-defined date formats are an advanced option, accessed from 2b. Identify input data sources > More > Other options.

#### **Fix**

A warning message was added for user-defined date strings that contain lower case elements for which upper case interpretations existing, such as 'm' versus 'M'.

### *Enhancement: Allow custom measurement units for temporal scale*

---

#### **Description**

When adding temporal support (scale) information for forecasts and observations, only measurement units available in the EVS library of units were supported. If the units were not available in the EVS units library, the support could not be defined.

#### **Cause**

Insufficient flexibility for user-defined measurement units in the tables of temporal support (scale) information for the forecasts and observations. The tables are accessed from 2b. Identify input data sources > More > Forecast scale and 2b. Identify input data sources > More > Observed scale.

**Fix**

Allowed custom attribute units to be defined for the temporal support (scale) information associated with the forecasts and observations (previously, this was constrained to units in the EVS unit library).

***Enhancement: Redundant computation of duplicate thresholds***

---

**Description**

When computing verification metrics for real-valued thresholds determined from climatological probabilities, those climatological probabilities may have selected duplicate real-valued thresholds. For example, when verifying precipitation forecasts, the probability of precipitation may be higher than several of the climatological probabilities requested. In that case, the verification results were computed for each duplicate threshold.

**Cause**

Failure to check for duplicate real-valued thresholds from the requested set of climatological probabilities before computing the verification results for those (possibly duplicate) thresholds.

**Fix**

Checks are now made for duplicate real-valued thresholds before computing the verification metrics. If duplicate thresholds exist, the verification results are only computed for the last of those thresholds (when arranged in order of magnitude). This avoids redundant calculations.

***Enhancement: Handling of large datasets at aggregated resolutions***

---

**Description**

When verifying large datasets for which the input data were at a higher temporal resolution than the required verification results, the EVS processed the input data in their native resolution before computing the aggregated pairs. In some cases, this resulted in an out-of-memory error because all input data were read in their native resolution before computing the aggregated pairs.

**Cause**

Insufficient options for determining how to process high-resolution input data for which verification results are required at a coarser resolution, thereby avoiding out-of-memory errors.

**Fix**

An option was added to the EVS project file to only store the verification pairs in their aggregated resolution required for verification. Also, adapted the routines for reading forecast data to aggregate the data from each file separately, avoiding the need to store the native data in memory prior to aggregating all data. This is only beneficial if the input data are stored in multiple files rather than one large file.

**Notes**

In order to circumvent the high memory usage by the EVS for those forecast locations where the forecasts are much more resolved (e.g. hourly) than the temporal resolution required for verification (e.g. daily), an option was included to store the paired data in their aggregate (e.g. daily) resolution

rather than their native resolution. Further, the forecast I/O was updated to allow for “on-the-fly” aggregation of the forecasts (i.e. after reading each file), thereby avoiding the need to read and store all forecasts at their native resolution. The option to store the verification pairs in their aggregated resolution is only available as an advanced option in the EVS project file. The option is accessed with the XML tag `<raw_pairs_in_aggregated_res>`, which should be true to store the raw pairs in their aggregated resolution. When using this option, any attempt to recompute verification results at a higher resolution than the resolution available in the aggregated pairs will result in the pairs being deleted and recomputed (possibly resulting in an out-of-memory error, unless sufficient memory is allocated to the reading of the observed and forecast data at their native resolution).

---

### *Enhancement: Use of NWS binary (“CS”) files for verifying observations*

---

#### **Description**

Reading of NWS binary files for “Conditional Simulations” (CS) was previously restricted to forecast data. It was not supported for verifying observations, yet the CS format may contain simulated streamflows that are useful for verification.

#### **Cause**

Inability to use the NWS binary (CS) format for verifying observations, which may comprise simulated streamflows.

#### **Fix**

Allowed the use of NWS binary (CS) files for verifying observations.

---

### *Enhancement: Inefficient sorting of very large datasets*

---

#### **Description**

When pairing very large datasets (comprising several gigabytes of data), the routine for sorting the pairs in trace order, i.e. by forecast valid time and lead time, became highly inefficient. The routine was optimized for small-to-medium-sized datasets. However, this optimization yielded negligible performance benefits for smaller datasets, while causing a bottleneck for very large datasets.

#### **Cause**

The routine used to sort verification pairs in trace order, i.e. by forecast valid time then lead time, was highly inefficient for large datasets.

#### **Fix**

Updated the routine used to sort verification pairs by forecast valid date then lead time to improve its efficiency for large datasets.

#### **Notes**

In practice, the new sort routine is marginally slower for small-to-medium sized datasets (i.e. up to several hundred MB), but the speed difference is on the order of fractions of a second, i.e. irrelevant. For large datasets (several GB), the previous routine would hang indefinitely and eventually result in an out-of-memory error. The new routine is able to sort large datasets in a few seconds. The

`sortByTrace` method of `PairedData.java` now uses a singly-layered `TreeMap` indexed with an appropriate object (`ValidDate.java`) to sort the verification pairs by forecast valid time and lead time. The `ValidDate.java` implements the `java.lang.Comparable` interface whose `compareTo` method implements the sort. Previously, a doubly-layered `TreeMap` was used with separate layers for the valid time and lead time.

### *Enhancement: Use of not-a-number (NaN) as “no data” values in the EVS*

---

#### ***Description***

The NaN identifier was allowed but not properly supported as the EVS “no data” identifier.

#### ***Cause***

While NaN was allowed as the “no data” identifier, it was not properly supported when testing for equality to NaN values (i.e. identifying the observed or forecast data as “no data”). In Java, testing for equality to NaN requires a separate routine, i.e. `Double.isNaN`. Any other test for equality will fail. The EVS allowed NaN values as the “no data” identifier, but did not implement the correct test for equality.

#### ***Fix***

The NaN identifier is no longer allowed as a “no data” identifier in the EVS. Upon attempting to use “NaN” or a non-finite value (negative or positive infinity) as the “no data” identifier, an error is thrown.

#### ***Notes***

The inability to use NaN as the “no data” identifier may be relaxed in future. However, this will require small updates to a large number of EVS source files that handle “no data” values. The handling of no data will need to be abstracted to a separate class and that class should implement proper handling of NaN as a “no data” identifier by checking for equality using `Double.isNaN` where NaN is used as the “no data” identifier, as well as other checks for equality to finite values.