

FREEWAT

Free and Open Source Software Tools for Water Resource Management

EU HORIZON 2020 Project

FREE and Open Source Software Tools for WATer Resource Management

FREEWAT User Manual

Volume 5

Observation Analysis Tool

by ***Massimiliano Cannata[(1)], Jakob Neumann[(1)], Mirko Cardoso[(1)]***

Version 0.3

December 5th, 2016

This version is confidential, only for FREEWAT Project Partners



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 642224

FOREWORD

FREEWAT is an HORIZON 2020 project financed by the EU Commission under the call WATER INNOVATION: BOOSTING ITS VALUE FOR EUROPE. FREEWAT main result is an open source and public domain GIS-integrated modeling environment for the simulation of water quantity and quality in surface water and groundwater with an integrated water management and planning module. Specific objectives of the FREEWAT project are: to coordinate previous EU and national funded research to integrate existing software modules for water management in a single environment into the GIS-based FREEWAT platform and to support the FREEWAT application in an innovative participatory approach, gathering technical staff and relevant stakeholders in designing scenarios for the proper application of water policies.
The open source characteristics of the platform allow to consider this an initiative "ad includendum", as further research institutions, private developers etc. may contribute to the platform development. Capabilities integrated in FREEWAT are:

- a dedicated module for water management and planning;
- a whole module for calibration, uncertainty and sensitivity analysis;
- a module for solute transport in the unsaturated zone;
- a module for crop growth and rural water management;
- a module dedicated to time-series processing to support advanced model calibration;
- tools for dealing with groundwater quality issues;
- tools for the analysis, interpretation and visualization of hydrogeological data.

FREEWAT Development has received funding from the following projects:

1. Hydrological part has been developed starting from a former project, named SID&GRID, funded by Regione Toscana through EU POR-FSE 2007-2013 (sidgrid.isti.cnr.it).
2. Porting of SID&GRID under QGIS has been performed through funds provided by Regione Toscana to Scuola Superiore S.Anna - Project Evoluzione del sistema open source SID&GRID di elaborazione dei dati geografici vettoriali e raster per il porting negli ambienti QGIS e Spatialite in uso presso la Regione Toscana (CIG: ZA50E4058A)
3. Saturated zone solute transport simulation capability has been developed within the EU FP7-ENV-2013-WATER-INNO-DEMO MARSOL. MARSOL project receives funding from the European Union's Seventh Framework Programme for Research, Technological Development and Demonstration under grant agreement n. 619120 (www.marsol.eu)
4. The previous codes have been integrated into FREEWAT under development within EU H2020 project FREEWAT - Free and Open Source Software Tools for Water Resource Management. FREEWAT project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement n. 642224 (www.freewat.eu)
This User Manual provides guidelines to properly install and use the FREEWAT plugin.

***Project Coordinator**:*
Rudy Rossetto, Istituto di Scienze della Vita – Scuola Superiore Sant'Anna, Via S. Cecilia 3, 56127 Pisa.
Capabilities of the FREEWAT platform were tested either by the developers group and by Project Partners. However, future applications of the platform may reveal errors and drawbacks not detected to date. Users are kindly requested to notify any error in both platform and User Manual. Please send this info to: rudy.rossetto@sssup.it

***To cite this User Manual**:*
Cannata, M., Neumann, j., Cardoso, M. FREEWAT User Manual, Volume 5 - Observation Analysis Tool. December 5th, 2016

***Text editing**:*
Giovanna De Filippis(2), Matteo Ghetta(2)

***Affiliation**:*
(1) Istituto di Scienze della Terra – SUPSI, Canobbio (CH)
(2) Istituto di Scienze della Vita – Scuola Superiore Sant'Anna, Pisa (IT)

Contents

Abstract iii

1 Introduction: Observation Analysis Tool 1
- 1.1 Dependancies . . . 1
- 1.2 Introduction to OAT library . . . 1
- 1.3 OAT data storage . . . 3

2 OAT menu 5

3 Add time-series 6
- 3.1 Interface overview . . . 6
- 3.2 Create sensor with data from CSV . . . 8
- 3.3 Create sensor with data from istSOS . . . 9
- 3.4 Create sensor with raw data . . . 10
- 3.5 Create sensor with a *MODFLOW list file* . . . 11
- 3.6 Create sensor from *MODFLOW Head Observation* . . . 12
- 3.7 Create sensor with *MODFLOW Gage* . . . 13

4 Process a time-series 14
- 4.1 Interface overview . . . 14
- 4.2 Applying a Method . . . 15
- 4.3 OAT *Method* library . . . 16

5 Manage sensors 17
- 5.1 The Manage Sensor Interface . . . 17

6 Compare sensors 19
- 6.1 The *Compare Sensors* interface . . . 19

7 OAT processing methods 20
- 7.1 Compare sensors . . . 20
- 7.2 Digital filters . . . 21
- 7.3 Exceedance probabilities . . . 22
- 7.4 fill . . . 22
- 7.5 Extract hydro events . . . 23
- 7.6 Hydrograph separation . . . 23
- 7.7 Calculate hydrologic indices . . . 23
- 7.8 Integrate . . . 24
- 7.9 Resample . . . 25
- 7.10 Data values . . . 25
- 7.11 Set quality . . . 26
- 7.12 Calculate statistics . . . 27
- 7.13 Subtract . . . 27

8 Creating a HOB layer using OAT sensors 28

References 30

Abstract

Observation Analysis Tool (OAT) is a FREEWAT module intended to provide users with enhanced time-series processing capabilities that are useful to support advanced model calibration.

It is designed to facilitate:

- the import of time series data into FREEWAT,
- the analysis of measurements, visualization and elaboration,
- the management of data,
- the usage of these data in FREEWAT model creation and calibration.

CHAPTER 1

Introduction: Observation Analysis Tool

This short introduction explains the basic concepts behind the Observation Analysis Tool (OAT) core library in order to let the user better understand the processes and better control the tool.

Dependancies

OAT depends on the following *Python* packages:

- **Pandas which depends on:**
  - NumPy: 1.7.1 or higher
  - python-dateutil: 1.5 or higher
  - pytz: needed for time zone support
- requests: needed to retrive data from web services
- isodate: needed to handle ISO 8601 datetime formats
- SciPy: needed to perform some of the implemented processes

Because OAT depends on other modules natively written in *C language* (*NumPy* and *ScyPy*), they need to be appropriately compiled on your PC, or a compatible pre-compiled version should be downloaded and installed.

Note: Different versions of QGIS may have installed different version of the libraries, so be aware of the version dependancies detailed above. Newest version of QGIS may already come with dependancies requirements satisfied.

Introduction to OAT library

The OAT library is composed of two main classes:

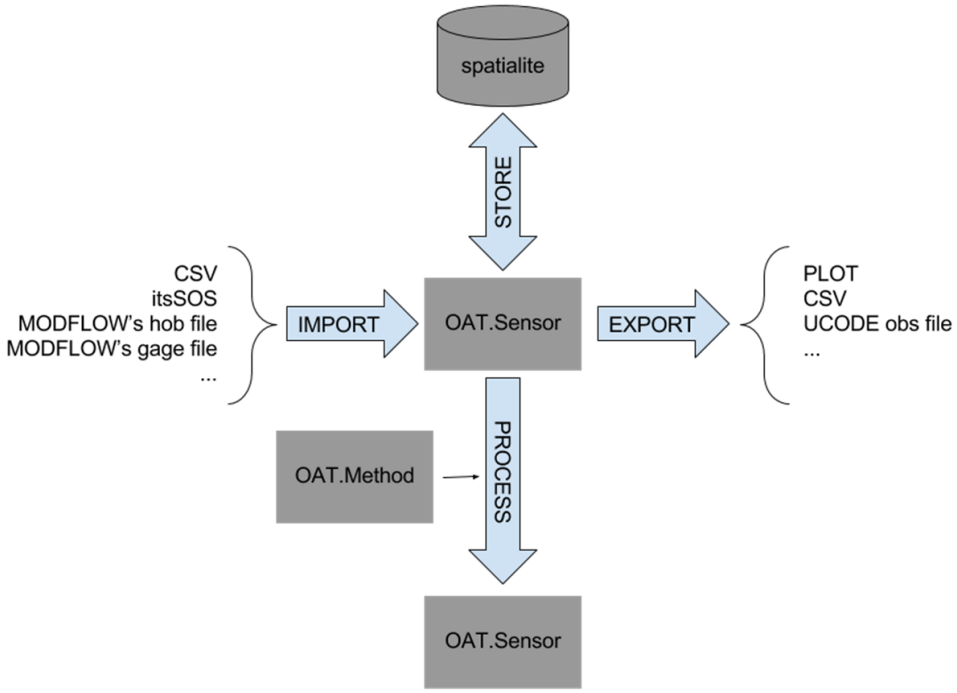
- *Sensor*
- *Method*

These two classes are at the base of the features offered by the FREEWAT's OAT tool. The *Sensor* is designated to handle time-series data and metadata while the *Method* class is designated to represent a processing method.

Classes structure

The OAT library applies the behavioral visitor pattern which allows the separation of an algorithm from the object on which it operates: thanks to this design pattern it is possible to add a new processing capability by simply extending the *Method* class without the need to modify the *Sensor* class.

The general structure and implemented usage of the library in FREEWAT is represented in the next figure:



In addition to requesting sensor data from an istSOS server (Cannata et al, 2015), OAT can retrieve data stored in local files or databases into the FREEWAT GIS environment for further use. MODFLOW model results (from list, HOB or gage files) can also be read. The *Sensor* object is stored in the spatialite database of the current FREEWAT project. A *Method* from the OAT library can be applied any *Sensor* object, which generally creates a new *Sensor* object. This new *Sensor* can be processed further with consequent *Methods* or saved in the database. *Sensor* objects can be plotted, exported as new CSV files or used to create UCODE observation input files.

Sensor object

The *Sensor* object is designed to represent a single point of observation (often realized by means of a sensor like a piezometer in the field) that is composed by a number of metadata and data.

The *Sensor* object has been concieved to be:

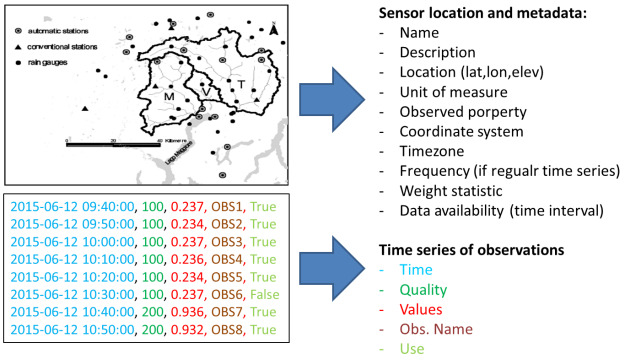
- created from existing formats;
- stored in permanent repository;
- exported in standard formats;
- processed for new sensor derivation.

Each *Sensor* object is characterized by a single time-series (one observed property only) that is represented by a data section consisting of a PANDA's time-series and a location/metadata section.

Sensor's metadata are used to identify the point of observation including the name, location (lat, lon, elev), observed property, unit of measure, coordinate system, time zone, frequency (if regular time-series), weight statistic and data availability (time interval). See *freewat_sensors* below.

Sensor data are composed by series representing:

- *time*: the time instant the data point is referring to;
- *quality*: a numerical index or value indicating the quality associated with the observation;
- *data*: the value of the observed/measured phenomena;
- *observation index*: a unique name assigned to each individual observation instant;
- *use*: a flag to indicate if an individual observation is to be used for further processing.



Method object

A *Method* object defines a process that can be applied to a *Sensor*.

Once a *Method* has been initiated it can be executed by a *Sensor* and the result can be used for further *Method* operations, detailed in the next '*OAT Method library*' section. The results of a process is generally a new OAT *Sensor*, so that processes can be concatenated, and the final resulting time-series can be saved in the FREEWAT model database or exported. Most of the currently available *Method* objects are based on TSPROC processing capabilities (Westenbroek et al. 2012).

See *OAT Method library* for further details on the *Methods*

OAT data storage

OAT requires an existing FREEWAT .sqlite database. If an attempt is made to add a time series to a QGIS project using OAT without a FREEWAT model table in the LP, an error message will be displayed. *sensor* objects created within FREEWAT, using any of the *Add time-series* options, will be saved within the current FREEWAT model database.

Sensor objects created within FREEWAT, using any of the *Add time-series* options, will be saved within the current FREEWAT model database.

Two types of objects are used to store the *sensor* data: 1. A virtual point layer containing the spation data for each *sensor* as well as the *sensor* metadata and 2. A data table for each created *sensor*

Sensor objects created within FREEWAT, using any of the *Add time-series* options, will be saved within the current FREEWAT model database.

Two types of objects are used to store the *Sensor* data:

1. A virtual point layer containing the spation data for each *Sensor* as well as the *Sensor* metadata and
2. A data table for each created *Sensor*

Upon the creation of the first *Sensor*, these two objects are created within the current FREEWAT database. A new data table is created and a feature is added to the point layer for every additional *Sensor* added to the project. The point layer is automatically loaded into the QGIS *Layers Panel* when the first OAT *Sensor* is created, as *Sensor object*.

Sensor layer

The Sensor layer is named *freewat_sensors* and is a geographical point layer with the following attributes:

- *id*: A unique progressing identifyer for each *Sensor* object;
- *name*: The name given to the *Sensor*;
- *desc*: A text discription of the *Sensor*;
- *tz*: An integer value representing the time zone in which the *Sensor* is located;
- *unit*: The unit of measurement of the property the sensor data represents;
- *prop*: The property that the sensor data represents;
- *freq*: The frequency with which the of the measurements are made, if they have a uniform frequency;
- *begin_pos*: The date and time of the first data point in the times series;
- *end_pos*: The date and time of the last data point in the times series;
- *statflag*: A descriptor for the type of statistics used in the quality of this *Sensor*;
- *use*: A flag to specify wether this *Sensor* is to be used for further processing, e.g. for the creation of head observations;
- *topscreen*: If the *Sensor* refers to an observation well, this indicates the top of the filtered screen in m a.s.l.;
- *bottomscreen*: If the *Sensor* refers to an observation well, this indicates the bottom of the filtered screen in m a.s.l.;

The *freewat_sensors* store the geomatries in the geographic spatial reference system WGS 84 and when added to your *QGIS Layers Panel* are then reprojected into the reference system of the current freewat project.

For each additional *Sensor* in the project, a new feature is added, corresponding to a new line to the attribute table of this point layer.

Data tables

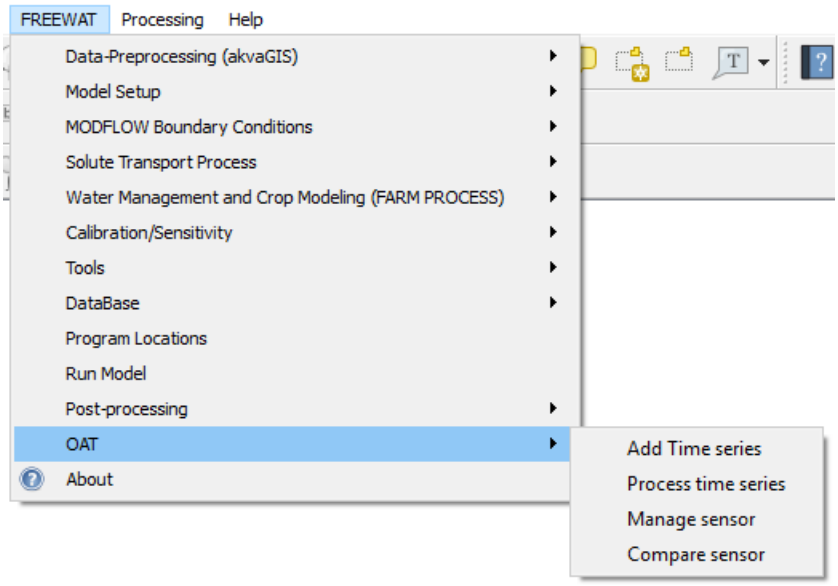
For each new *Sensor* added, a new data table is created in the FREEWAT database. This table has the same name as the *Sensor* it is associated with. Each table has the *time*, *quality*, *data*, *observation index* and *use* fields discribed above. The data tables are not loaded into the QGIS layers panel automatically. The data tables can be accessed either through the OAT oat_manage interface, or through the QGIS database manager.

CHAPTER 2

OAT menu

The OAT menu is composed by four entries:

- *Add Time Series*: used to create new time-series (sensors);
- *Process Time series*: used to perform time-series elaboration;
- *Manage sensor*: used to change and organize your sensor metadata;
- *Compare Sensor*: used to visually compare time-series in time.



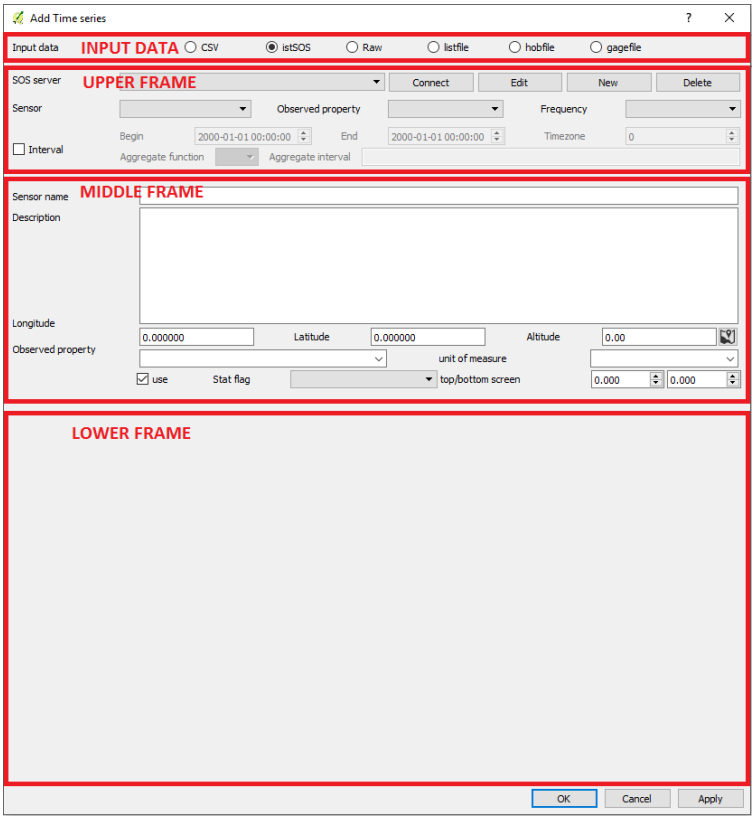
CHAPTER 3

Add time-series

Interface overview

When *Add Time series* is selected, a new dialog window is prompted.

This entry allows for creation of new *Sensor*.



The interface is subdivided in four frames: the input data frame and upper, middle and lower frames.

1. the *OK* button accepts these settings and saves the *Sensor* in the database;
2. the *Cancel* button close the window;
3. the *Apply* buttorn applies the settings of the upper and middle frames and creates a preview of the data imported with these settings in the lower frame.

Input data

The *Input data* option lets the user specify the data source type to be loaded. Depending on the option selected the upper and lower frame change their aspect.

Data can be loaded from local files, requested from online servers, or entered manually. In addition, MODFLOW model output data can be read from head observation files, the listing file and gage files.

Available options are:

- *CSV* (Comma separated values file);
- *istSOS* (an implementation of the Sensor Observation Service (SOS) standard from the Istituto scienze della Terra);
- *Raw* (Manually enter the values of a time series);
- *hobfile* (MODFLOW standard head observation output file, see MODFLOW HOB);
- *listfile* (MODFLOW listing file; here MODFLOW writes detailed run records including head and overall budget);
- *gagefile* (MODFLOW gage file, see MODFLOW GAGE);

The upper frame

Depending on the data source chosen in the *Input Data* frame, the upper frame will change its aspect.

The middle frame

The middle frame is is designated for specifying principal metadata:

- *Sensor name*: this is the name of this point of measure (unique within a FREEWAT model) - *MANDATORY*;
- *Description*: a textual description of the sensor - *OPTIONAL*;
- *Latitude*: the latitude in WGS84 (EPSG:4326) of the sensor - *MANDATORY*;
- *Longitude*: the longitude in WGS84 (EPSG:4326) of the sensor - *MANDATORY*;
- *Altitude*: the altitude in m a.s.l. of the sensor - *OPTIONAL*;
- *Observed property*: the physical property that is measured (An existing property van be chosen from the drop-down menu, or a new option can be typed manually) - *MANDATORY*;
- *Unit of measure*: the unit of measure of the observations (An existing property van be chosen from the drop-down menu, or a new option can be typed manually) - *MANDATORY*;
- *Use* option: used for MODFLOW observations, assigns the *Sensor* with a *use* field which can be used to indicates the include in future operations - *OPTIONAL*;
- *Stat flag*: used for MODFLOW observations, the type of statistic used to determine the weight of the observations - *OPTIONAL*;
- *Top/Bottom screen* used for MODFLOW observations, the used to define the screening of wells - *OPTIONAL*.

Note: The button at the end of the coordinates line allows for picking-up 2D coordinates from the QGIS map frame. Coordinates are automatically converted and stored in WGS84 whatever coordinate system is in use on your QGIS map frame.

The lower frame

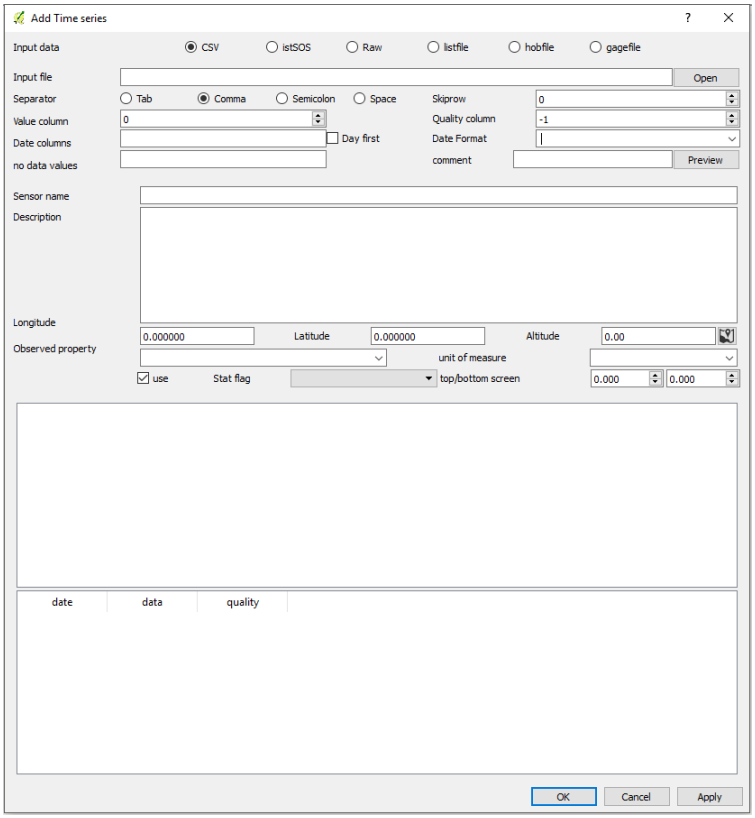
The lower frame displays a preview of the data for which the *Sensor* will be created. OAT uses matplotlib for plotting formatting and options.

Create sensor with data from CSV

The input file is selected through the *Open* button. According to the file selected, the correct separator, columns and date format must be defined.

Note that: * The column count starts at 0. * A drop down menu let's you select from several different *date formats*. * If the *date format* used in the CSV is not displayed here, it can be manually entered according to the Python strftime formatting.

In the *lower frame* a preview of the raw CSV file is shown. This can be used to correctly assign columns. When the *Preview* button is pressed a preview of the data as read with the set options is visualized in the lower row of the lower frame.



Create sensor with data from istSOS

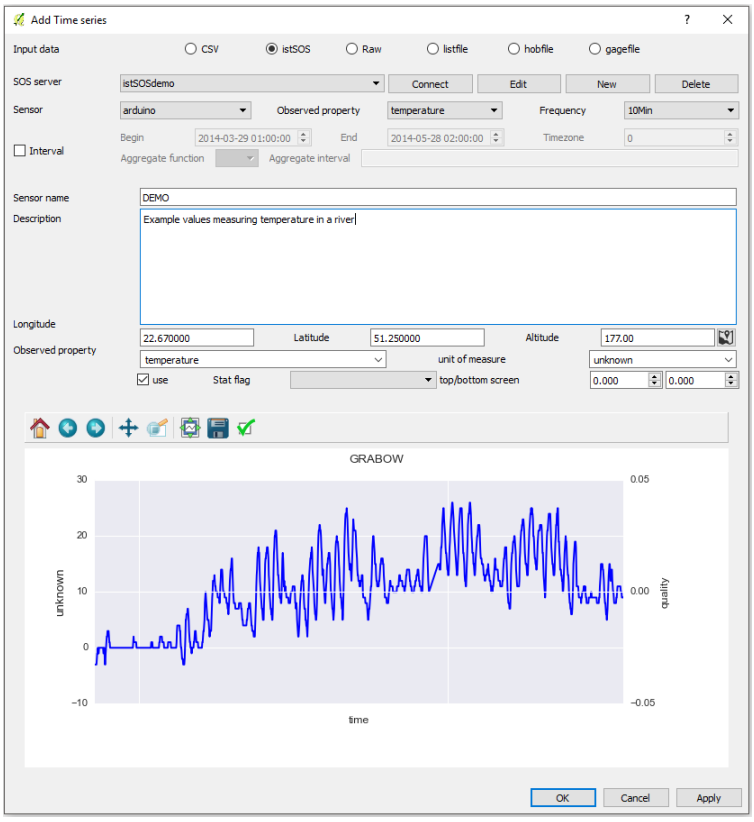
When you select the *istSOS* option as input data you have to connect to an istSOS server (similarly to database connection in QGIS you have to create a new one, or select an existing and connect). Servers can be selected from the *SOS server* drop-down menu. *Connect* creates the connection and displays the message:

```
Successfully connected to Server
```

If no server is available from the drop-down menu, a new connection can be established the the *New* button by specifying the server URL. Password protected URLs can also be accessed. Once connected, the *Sensor*, *Observed Property* and *Frequency* drop-down menu show the available options available from the selected server.

The *Interval* option allows to upload only data within a defined time interval set with a beginning and end date. If this option is selected, aggregate options become also available.

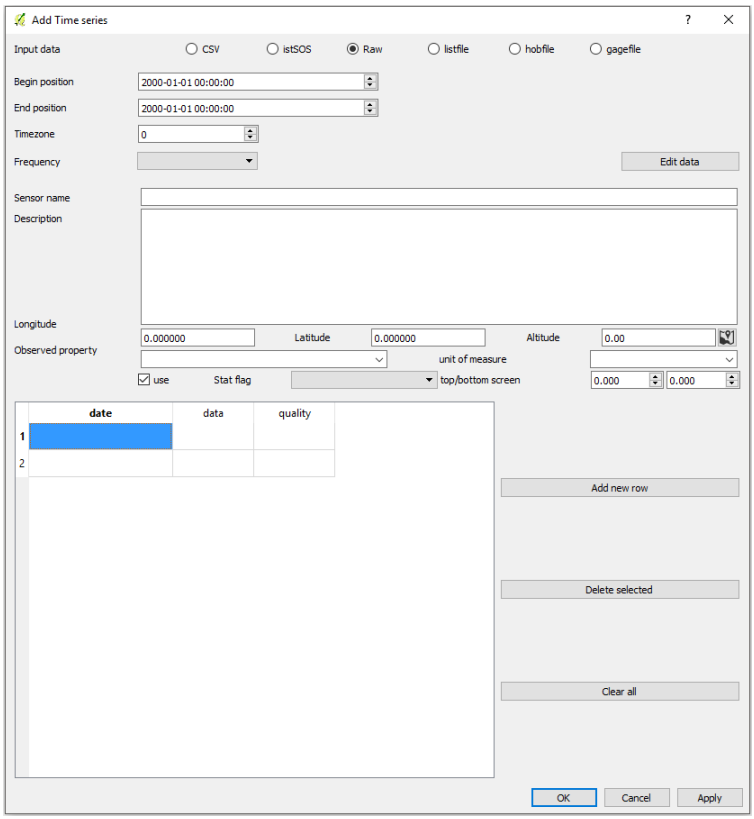
When loading data from an istSOS server, the middle frame is filled automatically with metadata from the server after the *Apply* button is pressed.



Create sensor with raw data

The *Raw* options can be used to enter data manually through the OAT.

The upper and middle frames specify metadata. In the lower frame the raw data can be entered row by row



When creating a *Sensor* from raw data the beginning and end date and time of the data to be entered must be specified. This is done through the *Begin position* and *End position* fields, with the option of specifying the *Timezone*. If the measurements were taken at a regular *Frequency*, this can be specified here.

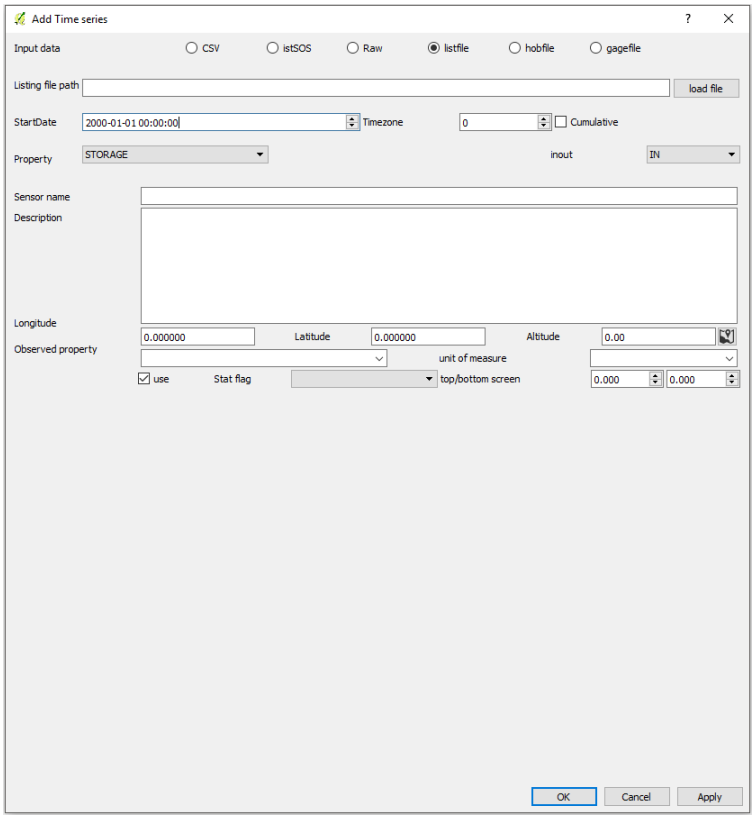
The *middle frame* is used to enter further metadata. See above.

To add a new data, the *Add new row* button must be used. It creates a new row in which the *date*, *data* and *quality* fields must be filled manually.

A row can be selected and deleted, and the entire data section can be cleared with the *clear all* button.

Create sensor with a *MODFLOW list file*

It is possible to create a time-series from a MODFLOW simulation trought the *MODFLOW listing file*.



The starting date should be set manually to match the starting date of the simulation. MODFLOW packages used in the simulation can be selected from the *Property* drop down menu. Selection of flows into or out of the model is specified through the *inout* field.

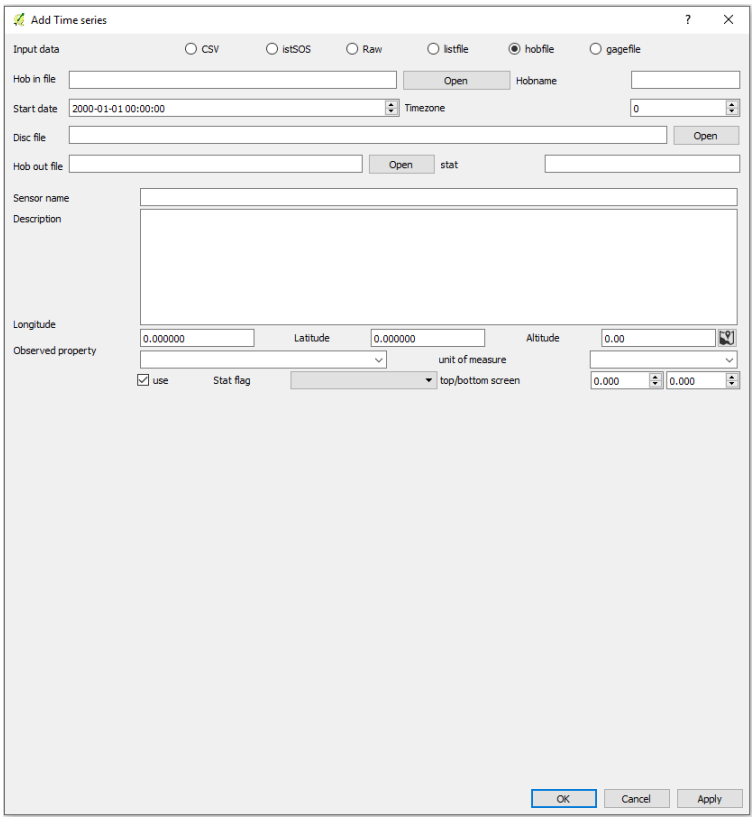
Note: Only packages included in the listing file can be read. If the selected property is not present, an error message will be displayed.

Listing file results can be displayed as rates per time step or as cumulative volumes through the *Cumulative* option.

The *Apply* button plots the listing file results for the selected property.

Create sensor from *MODFLOW Head Observation*

The simulated heads of a MODFLOW model can be loaded as a *Sensor* to create a new time-series. To be able to read the data from the simulation as a time-series, the specification of the MODFLOW *discretisation file* and *head input file* are required.



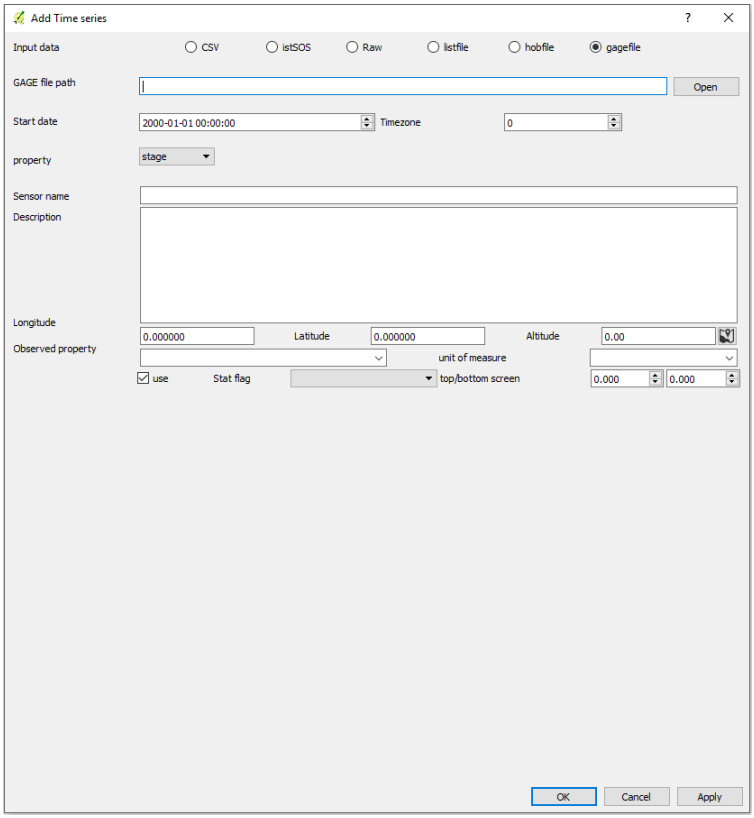
The user has to specify:

- *Hob in file*: the MODFLOW head observation input file (*.hob* files for FREEWAT projects);
- *Start date*: the starting date of the MODFLOW model;
- *Time Zone*: time zone modifier for starting date;
- *Disc file*: the MODFLOW standard file (*.dis* files for FREEWAT projects);
- *Hob out file*: the MODFLOW head results file (*.obh* files for FREEWAT projects);
- *Hobname*: the prefix name of the chosen observation point used in the head observation input file.

Note: The file extensions may differ from above if MODFLOW simulations are not created using FREEWAT

Create sensor with *MODFLOW Gage*

The MODFLOW gage file may contain model results for surface water bodies.



The results can be used to create a *Sensor* by selecting:

- *GAGE file path*: the location of the MODFLOW gage file (the file extension can vary);
- *Start date*: the starting date of the MODFLOW model;
- *Time Zone*: time zone modifier for starting date;
- *property*: *Gage file* property to be loaded.

Note: The available properties are set for the streamflow routing (SFR) and lake (LAK) packages; if the selected property is not in the gage file an error message will be siplayed when the *Apply* button is pressed.

CHAPTER 4

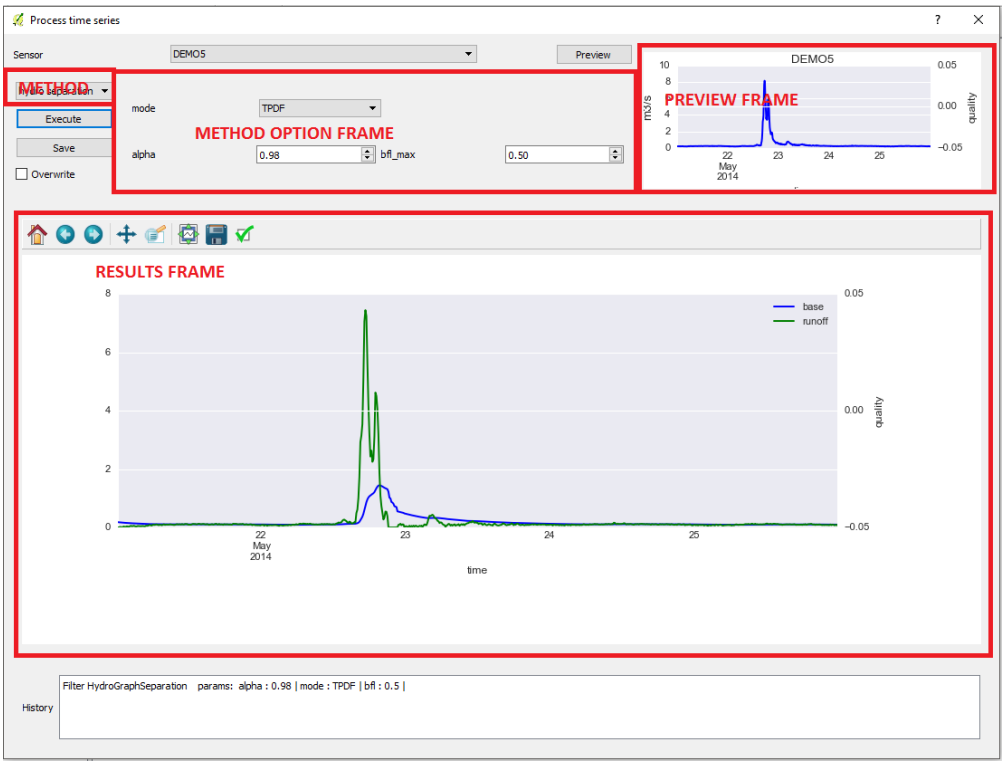
Process a time-series

Interface overview

When *Processing time-series* is selected, a new dialog window is prompted.

The window offers options to select *Sensor* and *Method*, apply them and save the results, as well as three frames which will change according to the options selected.

Additionally, a brief text history of the previously applied processes is displayed at the bottom of the window. The text displays the options used in the *Method Frame* and will change depending on the *Method* used.



Note: Unless the Overwrite option is selected, applying a *Method* to a *Sensor* does not overwrite the *Sensor* data. In this case a new temporary time-series is created, which may be saved as a *Sensor*.

The *Preview Frame*

The *Preview Frame* will show the plot of the currently selected time-series. If no time-series is displayed in the *Results Frame*, the selected *Method* will be applied to this time-series. By pressing the *Preview* button, the *Results Frame* will be cleared.

The *Method Option Frame*

Depending on the selected *Method* the *Method Option Frame* will change to display a mask of options specific to each *Method*. See *OAT Method library* for further details on available algorithms.

The *Results Frame*

The *Execute* button applies the selected *Method* using the options specified in the *Method option frame*. The *Results Frame* then shows the resulting time-series, which can then be saved as a new *Sensor*, or used for further processing.

Applying a Method

Once a *Sensor* has been selected, the following actions are required:

1. select the processing *Method* from the drop-down menu;
2. fill in the input parameters requested;
3. press the *Execute* button to perform teh calculations;

Users can use the results in one of the following way:

1. save the result in a new sensor (*Save* button);
2. overwrite the sensor time-series with the result (activate *Overwrite* option);
3. use the results for further processing (see *Concatenation* below).

Once a sensor has been loaded it is possible to select a *Method* to process the data and the appropriate input. The *Method Frame* will change according to the selected *Method*

The *Execute* button will run the *Method*, and when the execution is completed the resulting time-series is shown in the *Results Frame*.

If time-series are created it is always possible to save them specifying the new sensor names when requested. If more than one time-series are created by the applied *Method*, it is possible to save each seperately.

Concatenation

It is possible to apply several *Methods* to a *Sensor* consequtively, without having to save each time-series created. Once a time-series is loaded into the *Results Frame* any further *Method* will be directly applied to that time-series, not to the *Sensor* shown in the *Preview Frame*. The time-series shown in the *Results Frame* is not saved as a new *Sensor* until specified by the user and does not overwrite existing *Sensor* data.

OAT *Method* library

The OAT *Method* class has been designed so that it can be easily extended to include new processing capabilities.

An explanation of the currently supported *Methods* as well as the most up to date method library, can be found under: *OAT processing methods*

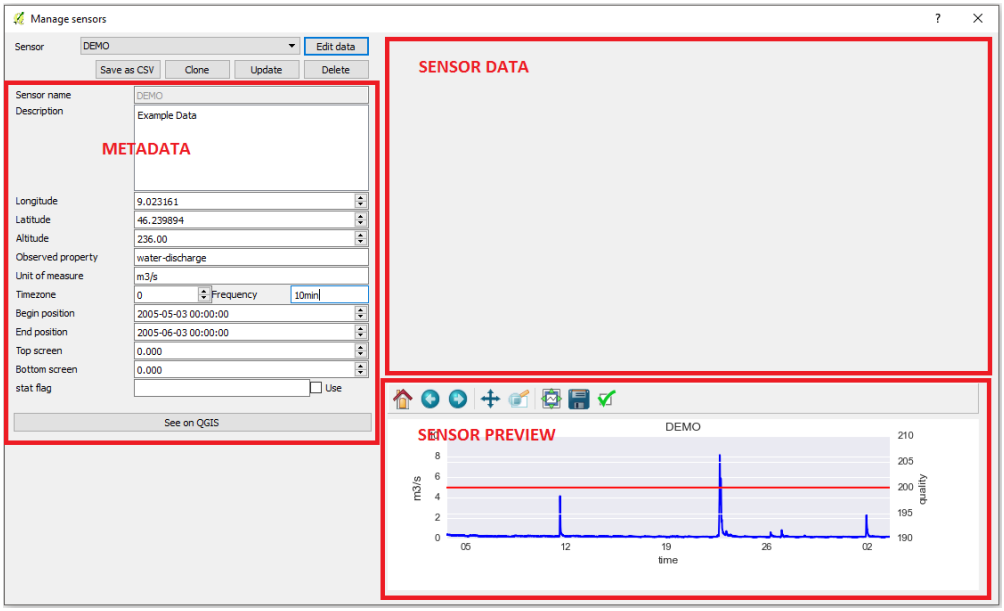
CHAPTER 5

Manage sensors

The Manage Sensor Interface

When *Manage Sensor* is selected, a new dialog window is prompted.

Through this interface is possible to manage and edit the *Sensor* saved in the database.



Once the desired sensor is selected in the drop-down list the metadata is automatically loaded and a preview of the data is shown at the bottom right window.

The *Manage Sensor* window is divided into three frames:

- *Metadata Frame*: displays the metadata for the selected *Sensor*;
- *Sensor Preview Frame*: an automatically loaded preview plot of the *Sensor* data;
- *Sensor Data Frame*: frame to display the data tables for the selected *Sensor*.

The *Edit Data* button opens the QGIS data table in the *Sensor Data Frame*. The table can be manually edited or manipulated using QGIS tools.

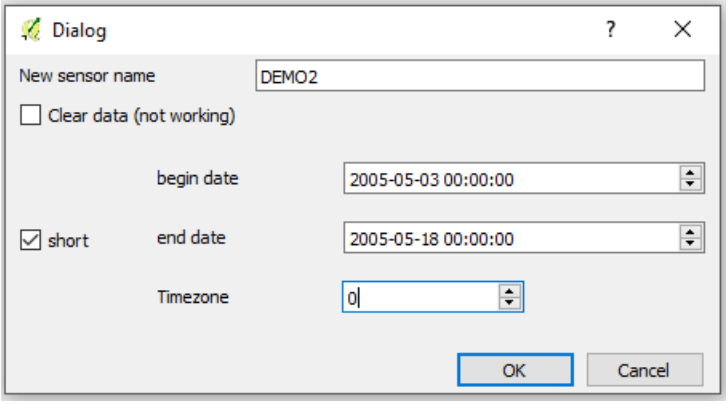
> **Note:** In older versions of QGIS, the data table opens as a new window. In QGIS 2.14, the data table is embedded into the *Manage Sensor* window. This different behavior is due to the different versions of the

graphical interface library Qt used.

Using the *Manage Sensor* it is possible to act on the following buttons:

- *Save as CSV*: export the sensor data as a new CSV file;
- *Update*: update the metadata, eventually modified, to reflect the *Metadata Frame* infomration;
- *Delete*: delete the selected sensor;
- *Clone*: create a copy of the selected sensor;
- *See on QGIS*: navigate to your QGIS map and zoom to the selected sensor.

Note: It is possible to refine the starting and ending dates for a cloned *Sensor* and use this option to filter the data in time.



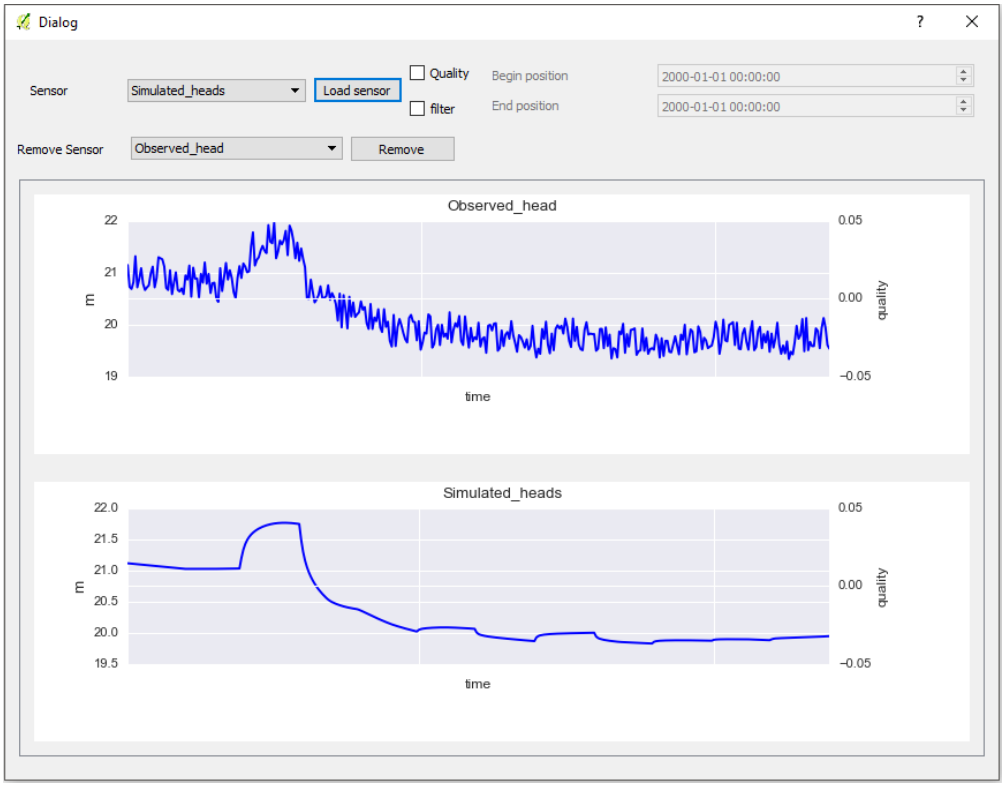
CHAPTER 6

Compare sensors

The *Compare Sensors* interface

When *Compare Sensors* is selected, a new dialog window is prompted.

Through this interface it is possible to plot several *Sensors* in the same window to visually compare them.



A number of sensors can be selected from the drop-down menu and added to the display window with the *Load sensor* button. If the *quality* check-box is selected only the *quality* data is loaded. If the *filter* option is checked the user have to specify a time interval for which sensor data should be loaded, e.g. when two *Sensor* to be compared do not have the same starting or ending dates.

Any *Sensor* loaded into the display can be removed by selecting it from the *Remove Sensor* drop-down menu and pressing the *Remove* button.

If required, axes and plotting options can be formatted using the inbuilt *matplotlib* tools.

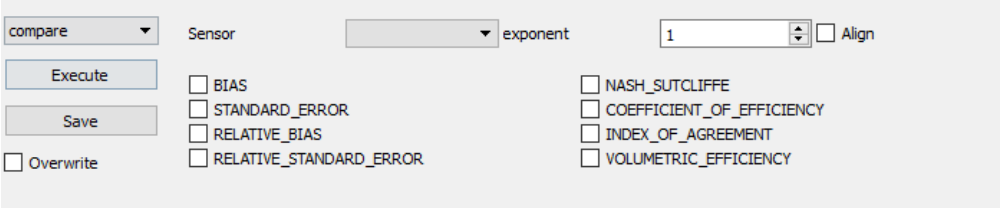
CHAPTER 7

OAT processing methods

This section gives an overview of all OAT methods currently implemented in the user interface.

Compare sensors

The *Compare Sensors Method* can be used to calculate several "Goodness of Fit" statistics for two *Sensors*.



Note: The observed time-series must be selected as the primary *Sensor* (next to the *Preview* button), and the simulated time-series should be selected as the *Sensor* in the *Method* mask

The options available are:

- Bias: the bias between a *observed* and *simulated* time-series.

$$B = \frac{1}{N}\sum(S_i - O_i)$$

- Standard error: the standard error between a *observed* and *simulated* time-series.

$$S = \sqrt{\frac{1}{N-1}\sum(S_i - O_i)^2}$$

- Relative Bias: the relative standard bias between the observed and simulated time-series.

$$B_r = \frac{B}{\overline{O}}$$

- Relative standard error: the relative standard error between the observed and simulated time-series.

$$S_r = \frac{S}{S_O}$$

- Nash Sutcliffe: calculates the Nash-Sutcliffe (1970) coefficient

$$R^2 = 1 - \frac{\sum (S_i - O_i)^2}{\sum O_i - \bar{O}}$$

- Coefficient of efficiency: calculates the coefficient of efficiency after Legates and McCabe (1999)

$$E_k = 1 - \frac{\sum |S_i - O_i|^k}{\sum |O_i - \bar{O}|^k}$$

- Index of agreement: calculates the index of agreement after Legates and McCabe (1999)

$$d_k = 1 - \frac{\sum |S_i - O_i|^k}{\sum (|S_i - \bar{O}| + |O_i - \bar{O}|)^k}$$

- Volumetric efficiency: calculates the "Volumetric Efficiency" proposed by Criss and Winston (2008)

$$VE = 1 - \frac{\sum |S_i - O_i|}{\sum O_i}$$

With:

$$\bar{O} = \frac{1}{N} \sum O_i$$

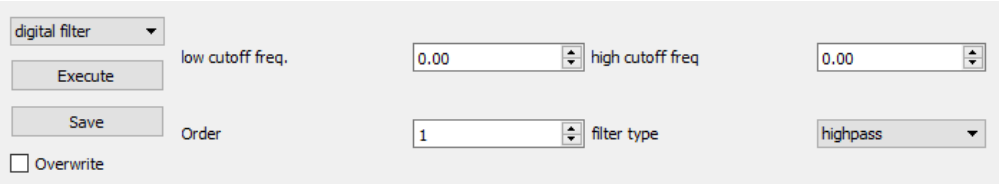
$$S_0 = \sqrt{\frac{1}{N-1} \sum (O_i - \hat{O})^2}$$

For all formulas presented above, S_i and O_i are the values of the simulated and observed time-series respectively. N is the number of terms in a series.

The *exponent* field represents the exponent ^{k} required to calculate the *coefficient of efficiency* and the *index of agreement*, with a value of 1 or 2.

Digital filters

The digital filter *Method* calculates a new *Sensor* by passing an existing *Sensor* through a digital filter to remove high or low frequency components. (low- and high-pass filters respectively). A high-pass filter removes long-term variations from a time-series, whereas a low-pass filter removes short term variations.



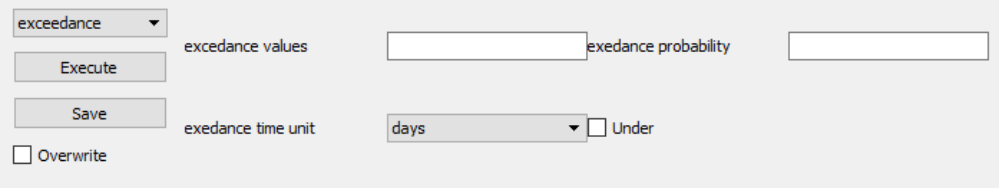
The user must specify:

- *low cutoff frequency*: the 3dB point of high frequency roll-off for a low-pass filter (in units of day $^{-1}$);
- *high cutoff frequency*: the 3dB point of low frequency roll-off for a high-pass filter (in units of day $^{-1}$);
- *Order*: the number of filter stages (integer between 1 and 3);
- *filter type*: highpass or lowpass.

Note: The cutoff frequencies must be be less than one-half the sample frequency of the *Sensor* to be filtered. E.g. the cutoff frequency for a *Sensor* with a 10 minute frequency must be less that 77 day^{-1}.

Exceedance probabilities

The *Exceedance probability method* calculates the exceedance probability after Searcy (1959) for a particular *Sensor*.



The user can specify the following:

- *excedance time unit*: the time unit in which the exceedance frequency will be displayed; allowed values are ['seconds','minutes','hours','days','years']; the default is 'days';
- *Under* checkbox: change the calculation to specify when values are *not* exceeded;

One of:

- *exceedance values*: a discharge value for which the exceedance probability will be calculated;
- *exceedance probability*: the discharge values that correspond to a given exceedance probability.;

fill



If a *Sensor* is missing data in the form of gaps, or if it contains no-data values, it can be filled using a variety of methods:

- *bfill*: backward fill (fills all the missing data values with the last data value before the no-data gap);
- *ffill*: forward fill (fills all the missing data values with the next data value after the no-data gap);

Note: If *bfill* or *ffill* are used, a *consecutive no data allowed* value is required to indicate the number of consecutive no-data values can be filled using the same value.

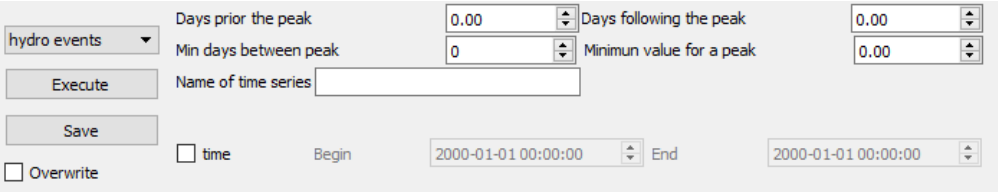
For more information on forward and backward fill methods , see pandas.DataFrame.fillna.

- *time*: interpolate proportional to time distance;
- *spline*: use spline interpolation;
- *linear*: linear interpolation;
- *quadratic*: quadratic interpolation;
- *cubic*: cubic interpolation.

For more information on interpolation fill methods , see pandas.DataFrame.interpolate.

Extract hydro events

The *extract hydrologic events method* can be used to extract hydrographs for a time period (in days) *preceding* and *following* a peak hydrologic event, such as a storm. The *minimum peak* value is to be defined, as well as the miniumum time between peaks. The *method* will produce a new time-series for every event with the *name of time-series* assigned.



The hydrologic events can be extracted from the whole time-series, or within time bounds selected though the *time* option.

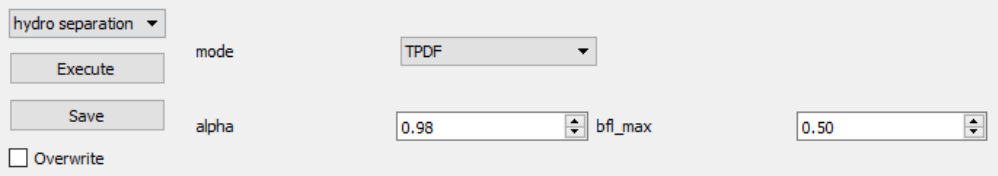
- *Days prior the peak*: the number of days prior to the peak to include in the event hydrograph;
- *Days following the peak*: the number of days following the peak to include in the event hydrograph;
- *Min days between peak*: minimum time between successive peaks, in days;
- *Minimum value for a peak*: minimum value for a peak;
- *Name of time-series*: suffix to be given to the created time-series; they will be named `seriesName+suffix+number` (e.g.: with a series named `TEST` and a suffix `_hyevent_N` we will have: [`TEST_hyevent_N1`, `TEST_hyevent_N2`, ...]);
- *time*: a tuple of two elements indicating the *Begin* and *End* of datetimes records to be used in peak extraction.

Hydrograph separation

The *hydrograph separation method* is a filter which produces two time-series, storm-flow and baseflow hydrograph, from stream discharge. Either the *Two Parameter Digital Filter* (Eckhardt, 2005) or *Single Parameter Digital Filter* (Nathan & McMahon, 1990) may be used. Both require the specification of rate of decay of baseflow relative to current flow rate, *alpha*. A value of 0.92 - 0.98 is recommended.

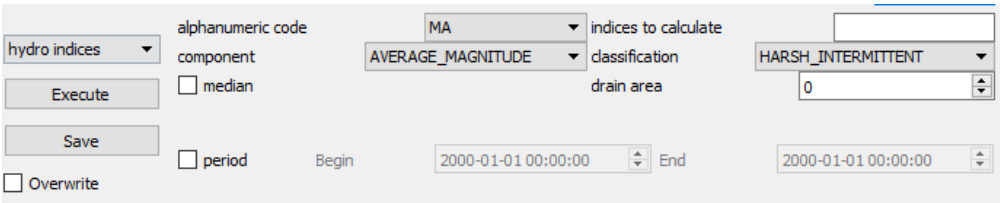
In case *TPDF* is used, the maximum value *bfl_max* of the base flow index (the long-term ratio of baseflow to total streamflow) is also required.

The resulting time-series may be saved as new *Sensor* objects.



Calculate hydrologic indices

The hydrologic indeces are a series of statistical measures of streamflow used to describe various ecologically important aspects of flow regime. Over 160 different hydrologic indeces can be calculated. For a complete discription of the available hydrologic indices, as well as the strings required to activate them, see the appendix in Olden & Poff (2003), Table 3-2 in Westenbroek et al (2012) or Henrickson (2006).



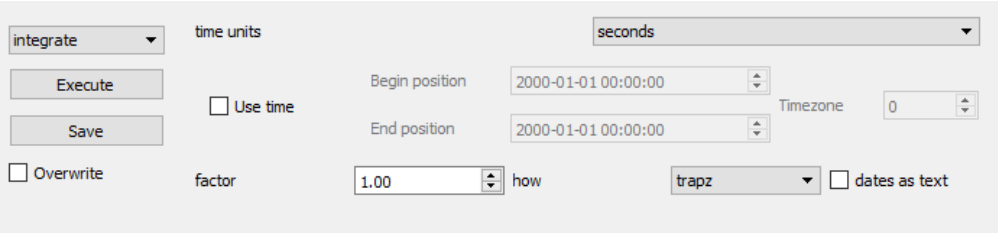
In order to specify the desired hydrologic index, the correct codes and keywords, taken from the sources above, must be specified in the mask. For some *hydrologic indices* the total area drained by a gaging station must be specified in m². The selected *hydrologic index* can be calculated on the whole time-series of the *Sensor* or for a user specified time-segment by selecting the *period* option and entering starting and ending dates.

The user must specify:

- *alphanumeric code*: one of [MA,ML,MH,FL,FH,DL,DH,TA,TL,TH,RA];
- *indices to calculate*: code that jointly with htype determines the indiced to calculate;
- *component*: used to specify the hydrologic regime as defined in Olden and Poff (2003). One of [`AVERAGE_MAGNITUDE`, `LOW_FLOW_MAGNITUDE`, `HIGH_FLOW_MAGNITUDE`, `LOW_FLOW_FREQUENCY`, `HIGH_FLOW_FREQUENCY`, `LOW_FLOW_DURATION`, `HIGH_FLOW_DURATION`, `TIMING`, `RATE_OF_CHANGE`]
- *classification*: Used to specify the hydrologic regime as defined in Olden and Poff (2003). One of [`HARSH_INTERMITTENT`, `FLASHY_INTERMITTENT`, `SNOWMELT_PERENNIAL`, `SNOW_RAIN_PERENNIAL`, `GROUNDWATER_PERENNIAL`, `FLASHY_PERENNIAL`, `ALL_STREAMS`];
- *median* requests that indices that normally report the mean of some other summary statistic to instead report the median value;
- *drain area*: The gauge area in m^2;
- *period*: a tuple of two elements indicating the Begin and End of datetimes records to be used.

Integrate

Several methods are available for integrating a *Sensors* time-series with respect to time.



The time-series can be integrated for the whole series, or within a lower and upper limit, specified by the *use time* option.

See SciPy documentation for further information on integration methods.

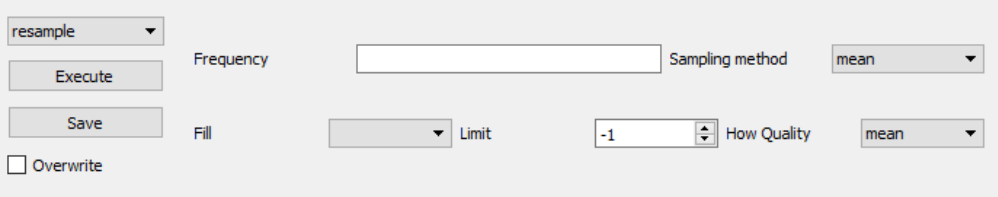
The user must specify:

- *time units*: the time units of data employed by the time-series, one of: 'seconds', 'minutes', 'hours', 'days', 'years';
- *use time*: a list of tuples with upper and lower time limits for volumes computation;

- *factor*: a factor by which integrated volumes or masses are multiplied before storage generally used for unit conversion (e.g.: 0.0283168 will convert cubic feets to cubic meters);
- *how*: the integration method, available methods are:
  - *trapz*: trapezoidal
  - *cumtrapz*: cumulative trapezoidal
  - *simps*: Simpson's rule
  - *romb*: Romberger's rule
- *dates as text*: defines if dates have to be returned as text (True) or Timestamp (False). Default is False.

Resample

The *resample method* calculates a new time-series with a given frequency by sampling values of a time-series with a different frequency.



The desired frequency for the new time-series is to be specified with the codes specified below, e.g. 6H to create a new time-series with data values every 6 hours. If no-data values exist in the time-series to be sampled, either forward filling or backward filling can be used (see: *fill*). In the same way, a limit for the number of consecutive allowable no-data must be specified (-1 means no limit).

If a time-series is resampled, the quality is resampled as well. The default sampling method for quality is the same as used for the data sampling method, unless specified.

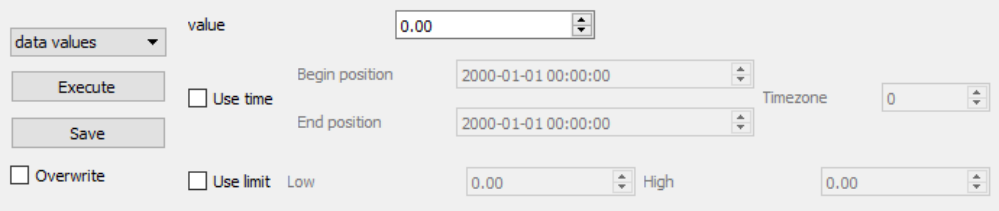
For more information on resampling methods, (see pandas.DataFrame.resample).

The user must specify:

- *Frequency*: an alphanumeric code specifying the desired frequency (A=year, M=month, W=week, D=day, H=hour, T=minute, S=second; e.g.: 1H10T);
- *Sampling method*: the sampling method ('mean', 'max', 'min', 'first', 'last', 'median', 'sum') for observations, the default is 'mean';
- *Fill*: if not null it defines the method for filling no-data ('bfill'= backward fill or 'ffill'=forward fill), the default is *None*;
- *Limit*: if not null it defines the maximum numbers of allowed consecutive no-data valuas to be filled;
- *How quality*: the sampling method ('mean', 'max', 'min', 'first', 'last', 'median', 'sum') for observation quality index (default is like 'sampling method').

Data values

The *data values method* can be used to set uniform data values for a *Sensor* within time bounds, or to remove values that fall outside of specified value bounds.



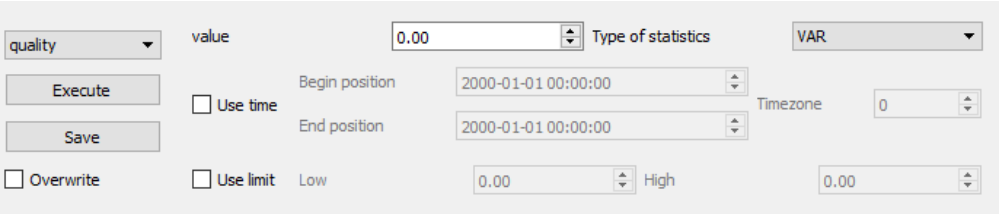
The user must specify:

- *value*: the value to be assigned;
- *Use time*: a flag to determine upper and lower time limits for assignment. The bounds are closed bounds (t0 >= t <= t1);
- *Use limit*: a flag to determine upper and lower value limits for assignment. The bounds are closed bounds (min >= x <= max) e.g: [(None,0.2),(0.5,1.5),(11,None)] will apply: if data is lower then 0.2; –> (None,0.2); or data is between 0.5 and 1.5 –> (0.5,1.5); or data is higher then 11 –> (11,None).

Set quality

The *set quality method* can be used to set uniform *quality* values for a *Sensor* within time bounds, or to remove values that fall outside of specified value bounds.

Note: The quality field in the data tables can also be used to represent a quality index, as well as a value used to calculate the weight of a data point. This method assumes that the latter is the case. The *set quality method* allows the user to specify the process through which weights will be calculated through the *type of statistic* drop down menu, e.g. in UCODE. In this case *statistic* refers to the value in the *quality* field of a particular data index.

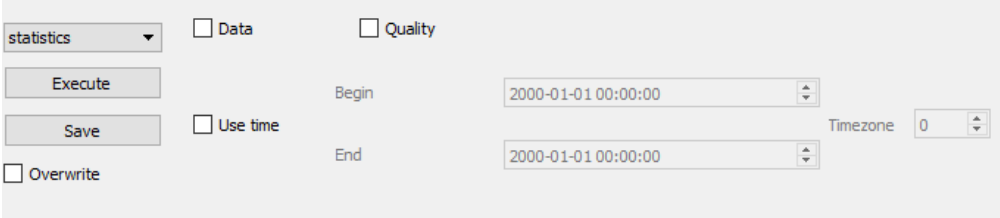


The user must specify:

- *value*: the value of the weight to be assigned;
- *Type of statistic*: the type of statistics the weight is estimated from, accepted values are:
  - VAR (Variance): the observation weight is calculated as $1/Statistic$.
  - SD (Standard deviation): the observation weight is calculated as $1/(Statistic)^2$;
  - CV (Coefficient of variation): the observation weight is calculated as $1/(Statistic * ObsValue)^2$.
  - WT (Weight): the observation weight is equal to $Statistic$;
  - SQRWT (Square root of the weight): the observation weight is calculated as $Statistic^2$;
- *Use limit*: A flag to determine upper and lower value limits for weigth assignment. The bounds are closed bounds (min >= x <= max) e.g: [(None,0.2),(0.5,1.5),(11,None)] will apply: if data is lower then 0.2 –> (None,0.2) or data is between 0.5 and 1.5 –> (0.5,1.5) or data is higher then 11 –> (11,None).
- *Use time*: Used when setting upper and lower time limits for weight assignment. The bounds are closed bounds (t0 >= t <= t1).

Calculate statistics

The *calculate statistics method* returns some basic satistics for the *Sensor* time-series. The statistics can be performed for a selected time period and on the data values as well as their associated quality values.



The statistics calculated are:

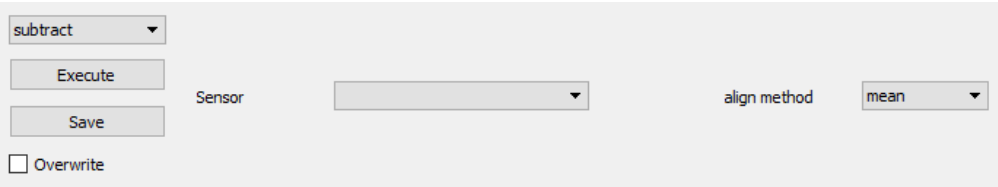
- *count*: returns the number of data values in the series;
- *std*: the standard deviation of the series;
- *min*: the minimum value in the series;
- *max*: the maximum value in the series;
- *50%*: the median value in the series;
- *25%*: the first quartile value in the series;
- *75%*: the third quartile value in the series;
- *mean*: the mean of the series.

The user must specify:

- *Data*: a flag to determinewether to compute statistics of data (default is True);
- *Quality*: a flag to determine wether to compute statistics of quality (default is False);
- *Use Time*: used when setting upper and lower time limits for statistic calculation. The bounds are closed bounds (t0 >= t <= t1).

Subtract

The *subtract method* can be used to subtract the data values of a *Sensor* from the data values of the selected *Sensor*.



If not aligned, data values of the two time-series are aligned using the mean with respect to the first term in each series.

CHAPTER 8

Creating a HOB layer using OAT sensors

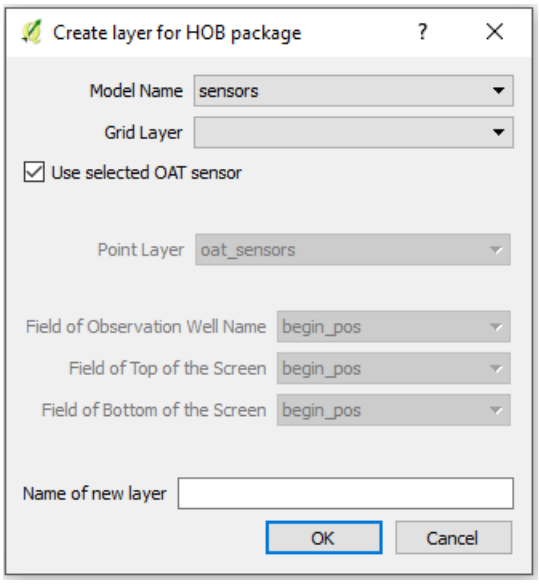
FREEWAT incorporates model calibration using UCODE. It is possible to use OAT *Sensors* to create a head observation layer for model calibration within FREEWAT. Knowledge of MODFLOW observations and UCODE is suggested for using this feature. In order to use this feature the relevant *Sensor* objects need to have the following metadata fields filled:

- *statflag*: a descriptor for the type of statistics used in the quality of this *Sensor*. Accepted values are (See UCODE for further information):
  - VAR (Variance): the observation weight is calculated as $1/Statistic$.
  - SD (Standard deviation): the observation weight is calculated as $1/(Statistic)^2$;
  - CV (Coefficient of variation): the observation weight is calculated as $1/(Statistic * ObsValue)^2$.
  - WT (Weight): the observation weight is equal to $Statistic$;
  - SQRWT (Square root of the weight): the observation weight is calculated as $Statistic^2$;
- *use*: a flag to specify wether this *Sensor* is to be used for further processing;
  - this must be TRUE for any sensors used to create a head observation layer;
- *topscreen*: the top of the filter screen of the observation well in m a.s.l.
- *bottomscreen*: the bottom of the filter screen of the observation well in m a.s.l..

The data table for every *Sensor* to be included will use the *quality* field to specify the **statistic** value which will be used with the *statflag* methods to determine observation weight (See above).

Note: The *initial date* and *initial time* fields of the FREEWAT *modeltable* will be used to assign the observations of the *Sensors* to the correct stress period and time step using the model *timetable*. These fields need to be assigned correctly to assure correct alignment of model time.

If the data in the *freewat_sensors* and the data table for each head observation conforms to the requirements, a new HOB layer can be creating using the FREEWAT interface. The head observation points must be selected in the QGIS map window. Once the head observation points have been selected, the user can open the *Create Head Observation Layer* dialog window from the FREEWAT *Calibration/Sensitivity* menu.



In this dialog window, a check-box allows the user to *Use selected OAT sensor*. Once this option has been selected, the other options in the dialog, apart from *Grid Layer* and *Name of new Layer* become locked. By specifying a unique name, FREEWAT will read the *freewat_sensor* metadata and the corresponding data tables to create a new HOB layer from the **selected** OAT *Sensors*. Sensors not selected in QGIS will not be used.

The OAT created HOB layer can be used for model calibration in FREEWAT. See FREEWAT manual Vol. 6 for further information.

References

Cannata M, Antonovic M, Molinari M, Pozzoni M (2015). istSOS, a new sensor observation management system: software architecture and a real-case application for flood protection. Geomatics, Natural Hazards and Risk, 6(8), 635-650.

Criss RE, Winston WE (2008). Do Nash values have value? Discussion and alternate proposals. Hydrological Processes, v. 22, no. 14, p. 2723–2725.

Eckhardt K (2005). How to construct recursive digital filters for baseflow separation. Hydrological Processes, v. 19, no. 2, p. 507–515

Harbaugh AW (2005). MODFLOW-2005, The U.S. Geological Survey Modular Ground-Water Model - the Ground-Water Flow Process. U.S. Geological Survey, Techniques and Methods 6–A16, 253 p.

Henricksen JA, Heasley J, Kennen JG, Nieswand S (2006). Users' manual for the Hydroecological Integrity Assessment Process software (including the New Jersey Assessment Tools): U.S. Geological Survey Open-File Report 2006–1093, 72 p.

Hill MC, Banta ER, Harbaugh AW, Anderman ER (2000). MODFLOW-2000, the U.S. Geological Survey modular ground-water model - User guide to the Observation, Sensitivity, and Parameter-Estimation Processes and three post-processing programs. U.S. Geological Survey, Open-File Report 00-184, 210 p.

Hunter JD (2007). Matplotlib: A 2D Graphics Environment, Computing in Science & Engineering, 9, 90-95.

Legates DR, McCabe GJ (1999). Evaluating the use of "goodness-of-fit" measures in hydrologic and hydroclimatic model validation. Water Resources Research, v. 35, no. 1, p. 233–241.

McKinney W (2010). Data Structures for Statistical Computing in Python, Proceedings of the 9th Python in Science Conference, 51-56

Nash JE, Sutcliffe JV (1970). River flow forecasting through conceptual models part I—a discussion of principles. Journal of Hydrology, v. 10, no. 3, p. 282–290.

Nathan RJ, McMahon TA (1990). Evaluation of automated techniques for base flow and recession analyses. Water Resources Research, v. 26, no. 7, p. 1465–1473.

Olden JD, Poff NL (2003). Redundancy and the choice of hydrologic indices for characterizing streamflow regimes: River Research and Applications, v. 19, no. 2, p. 101–121.

Poeter EP, Hill MC, Lu D, Tiedeman CR, Mehl S (2014). UCODE_2014, with new capabilities to define parameters unique to predictions, calculate weights using simulated values, estimate parameters with SVD, evaluate uncertainty with MCMC, and More. Integrated Groundwater Modeling Center Report Number: GWMI 2014-02.

Searcy JK (1959). Flow-duration curves, Manual of hydrology, pt. 2, Low-flow techniques. U.S. Geological Survey Water-Supply Paper 1542–A, 32 p.

van der Walt S, Colbert SC, Varoquaux G (2011). The NumPy Array: A Structure for Efficient Numerical Computation, Computing in Science & Engineering, 13, 22-30.

Westenbroek SM, Doherty JE, WalkerJF, Kelson VA, Hunt RJ, Cera TB (2012). Approaches in highly parameterized inversion: TSPROC, a general time-series processor to assist in model calibration and result summarization. U.S. Geological Survey Techniques and Methods, (7-C7), 112.