


# 03 Display Groups

What	DisplayGroups.xml
Config group	SystemConfigFiles
Required	no
Description	Defines pre-configured displays
schema location	<a href="https://fewsdocs.deltares.nl/schemas/version1.0/displayGroups.xsd">https://fewsdocs.deltares.nl/schemas/version1.0/displayGroups.xsd</a>

 Still to document: TimeSpanComplexType

- Introduction displayGroups
  - plot
  - displayGroup
- subplot (plot)
  - subPlotType
  - leftAxis and rightAxis
  - inverted
  - logarithmic
  - drawingOrderInverted
  - thresholdAxis
  - subplot types
    - line
    - area
      - filled vertical lines (area)
    - clusteredBars
    - variableLineWidth
  - line elements
    - axis (dual y-axis plot)
      - Multiple TimeSeries plots (dual y-axis plot)
      - Stage/discharge plots (dual y-axis plot)
  - timeSeriesSet
    - chainageLocationSetId
    - domainParameterId
      - Single Domain Axis Time Series / 1D Spectra plots
      - Dual Domain Axis Time Series / 2D Polar Spectra plots
    - visibleInTable, visibleInPlot
  - descriptiveFunctions
  - additionalTimeSeries
- locationLoop (displayGroup)
- display (displayGroup)
  - startForwardLookingPeriod
  - tickTimeStep
- singleLocationDisplays (displayGroup)
  - showAsVerticalProfile
  - showAsScatterPlot
    - Pairing of time series
  - statisticalFunctionId

## Introduction displayGroups

A list of pre-configured displays can be configured in the display groups. When available on the file system, the name of the XML file is for example: DisplayGroups.xml

The pre-configured displays are organised in a tree view in the time series display (see example in Figure 1). Each pre-configured display is identified by its name, and may include one or more subplots, each with one or more time series lines.

Another option is to plot a longitudinal profile in the time series display (see figure 2). The main difference with the normal time series plot is, that on the X-axis the river chainage is plotted. With the control toolbar a specific time step can be selected.

displayGroups schema: A display group forms one of the main nodes in the tree view and may contain multiple displays. Multiple display groups may be defined.

The display groups are configured in two parts: plot and displayGroup

Elements:

- **description**: Optional description
- **plot**: Plot complex type
- **displayGroup**: displayGroup complex type

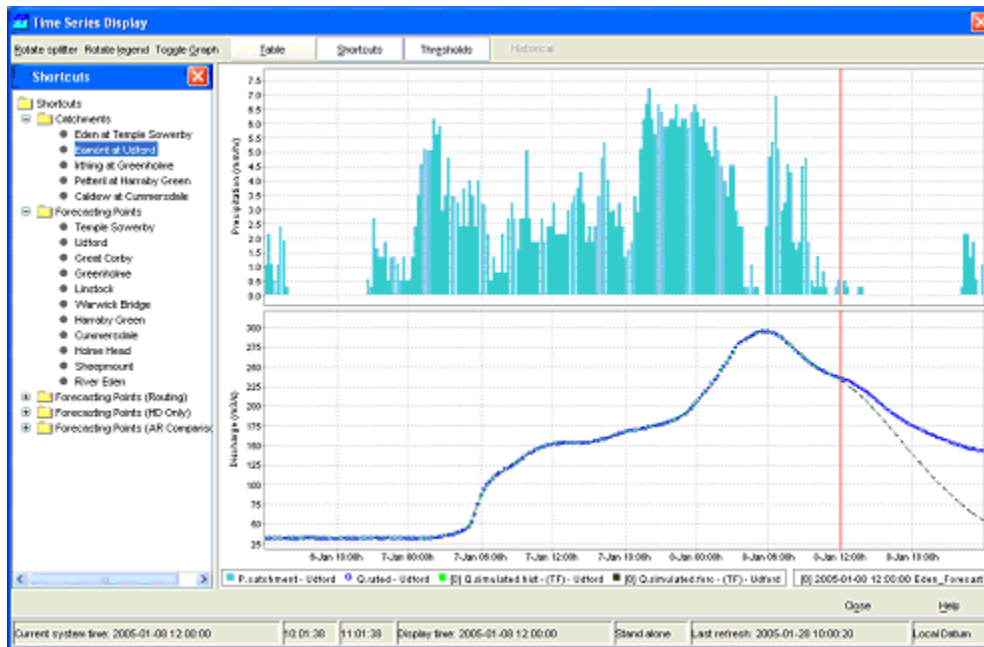


Figure 1 Left: Example of time series display, showing two sub-plots and tree-view of pre-configured displays  
 Figure 2 Right: Example of time series display, showing longitudinal profile.

## plot

Defines the plots relating to the displayGroup, for example in the Fractions displayGroup a stackPlot is defined with a max and min (this file is attached as an example).

Each displayGroup gets assigned a plotId, which links to the definitions of the plots and the time series set to be used.

Please note that the colours, linestyle, precision etc are defined in the [TimeSeriesDisplayConfig](#).

- Attribute: id
- Elements:
  - description
  - legendFontSize: Font size of legend
  - axis TitleFontSize: Title font size
  - tickLabelFontSize: tick label font size

- **tickTimeStep**: to tie ticks to a time step (since 2018.02)
- **synchronizeParameterAxis**: synchronize the axis range of subplots displaying time series for the same parameter group (since 2018.02)
- **visibilityGroup**: defines order of the visibility groups in the drop down box (since 2019.02)
- **subplot**
- **plotViewerId** & **tableViewerId**: to use alternative viewers for plot or tables

## displayGroup

Defines the groups of plots to be viewed (i.e. the branches of the shortcuts in the display), for example "Rain gauges", "Gauges" and "Fractions" in figure 3 below.

The names of the subplots can be added (e.g. "MacRitchie" and "Woodleigh" below).

- Attributes: **name** and **id**
- Elements:
  - **description**: Optional description
  - **nodeId**
  - **viewPermission**
  - **editPermission**
  - **locationLoop**
  - types of display
    - **display**
    - **singleLocationDisplays** Single location displays can be used to easily create plots for individual locations. This can either be done for individual locations, or alternatively for a locationSet which will then produce a list of location with their corresponding plot.
    - **singleParentLocationDisplays** Adds multiple displays at once, each showing the children and parent for one parent location.
    - **displayGroupId** (since 2023.01) include a displayGroup defined elsewhere by referencing the id.

displayGroups

version

1.0

xmlns

http://www.wildelft.nl/fews

xmlns:xsi

http://www.w3.org/2001/XMLSchema-instance

xsi:schemaLocation

http://www.wildelft.nl/fews http://fews.wildelft.nl/schemas/version1.0/displayGroups.xsd

plot (3)

id

subplot

1

Fcast

subplot (2)

2

RainGauges

subplot (1)

timeSeriesSet

1

timeSeriesSet (2)

moduleInstancId

valueType

parameterId

locationSetId

timeSeriesType

timeStep

relativeViewPeriod

readWriteMode

1

ImportPubRts

scalar

P.obs

MetGauges\_P.obs

external historical

timeStep unit=hour

relativeViewPeriod

add originals

2

ImportPubRts

scalar

P.obs

MetGauges\_P.obs

external historical

timeStep unit=day

relativeViewPeriod

add originals

3

Fractions

subplot (1)

stackPlot

max

min

timeSerie...

1

true

1

0

timeSerie...

displayGroup (3)

name

display

1

Rain gauges

display (2)

2

Gauges

displayGroup

3

Fractions

display (2)

name

relativeViewPeriod

locationId

plotId

1

MacRitchie

relativeViewPeriod

MacRitchie

RainGauges

2

Woodleigh

relativeViewPeriod

Woodleigh

RainGauges

name

relativeViewPeriod

locationId

plotId

1

Kallang Basin

relativeViewPeriod

KB

Fractions

2

Marina Bay

relativeViewPeriod

MB

Fractions

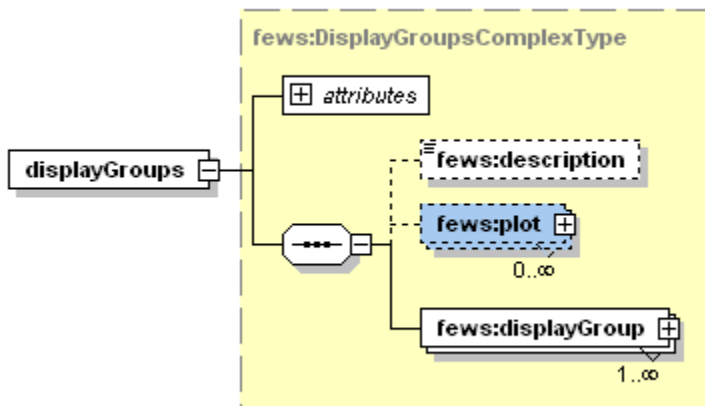


Figure 3 - example of a configured displayGroup file (click to enlarge)

Figure 4 Root element of the display groups definition



Display groups may be defined while DELFT-FEWS is running and reloaded by re-opening the time series dialogue. If a mistake is made, then the shortcuts item to open the tree view will not appear and an appropriate message will be generated. After resolving the mistake the item will again become available on re-loading the display.

## subplot (plot)

Element of plot. Root element for each subplot. Multiple sub-plots may be defined per plot.

Attributes:

- **min** (max is not implemented): set default minimum value for y-axis, which will be overruled by data that is below this value
- **yAxisScalingType** (since 2020.02): Default is `scaleOutsideMinAndMax`. If it is set to `scaleOutsideMinAndMax`, the y axis will scale to the data available. If it is set to `fixedBetweenMinAndMax`, the min and max of the y axis will always be the configured values, regardless of the data. This could make the entire data set fall of the visible chart. If all the data is in the middle of min and max withing a small range, the axis will scale to the data, it will appear as a small line in the middle. If it is set to `scaleBetweenMinAndMax`, the min will never be smaller than what is configured, and the max will never be greater than configured, but if the available data falls between these values, the axis will scale so that the data fills the whole area. If the data falls entirely outside of min and max, it will not be visible.
- **convertDatum**
- **stackPlot**—deprecated since 2014.02

Elements :

- **description**
- **subPlotType**: `stack`, `colorMapBars`, `horizontalColorCode`
- **axisLabel**, **axisScaleUnit**, **lowerMarginPercentage**, **upperMarginPercentage**
- **leftAxis**, **rightAxis**
- **barMarginPercentage** (Since 2018.02) Percentage of the bar width by which it will be reduced, to create space between the bars. General option to be applied to all plots containing bars (e.g. "horizontalColorCode" or bar charts). Overrules the general `barMarginPercentage` for a specific subplot. Default value is the `barMarginPercentage` of the [TimeSeriesDisplayConfig.xml](#) (which defaults to 0).
- **inverted**
- **logarithmic**
- **drawingOrderInverted**
- **splitForDifferentRuns**
- **plotweight**
- **plotSeparatorWeight**
- **thresholdAxis**
- **thresholdAxisScaling**
- **forecastConfidenceTimeSpan**
- sub plot types
  - **line**
  - **area** Fills the area between multiple time series.
  - **clusteredBars**
  - **variableLineWidth**
- **classBreaksId**, **classBreaks**
- **direction**
- line elements
  - **color**: Overrides colors specified in the `timeseriesdisplay`
  - **lineStyle**: Line style of time series marker line. Enumeration of "solid", "none", "bar", "dashdot", "dashed", "dotted". When a suffix of "; thick" is given, 1 will be added to the linewidth found in the `TimeSeriesDisplayConfig.xml`. For example, if a linewidth of 2 is configured in the `TimeSeriesDisplayConfig` and for a subplot the `linestyle` is set to "solid;thick", the resulting linewidth will be 2 + 1 = 3.
  - **markerStyle**, **markerText**, **markerSize**
  - **axis** for dual axis plots
  - **ratingAxis**, **lookupAxis**, **datumAxis**
  - **visibleInLegend**
  - **label**
  - **visibleInTable**, **visibleInPlot**
- **descriptiveFunction**, with **value** and **timeSpan**
- **timeSeriesSet**, with
  - **chainage location set**,
  - **single domain axis** and
  - **dual domain axis**
- **additionalTimeSeries**

## subPlotType

By default the `<subPlotType>` is set to 'standard', which means the default plot type for the configured `timeSeriesType` will be used. Optionally, you can configure the following other subplot types.

- **stack** Time series of this subplot are plotted as stacked areas (if `lineStyle` is solid) or stacked bars (if `lineStyle` is bar).
- **colorMapBars** Time series of this subplot are plotted as stacked bars. Color of the bar matches the paint scale color associated with the bar value. Height of all bars is the same, and equals Y-axis range / number of series in the plot.
- **horizontalColorCode**

A horizontal bar is plotted for each time series. The color of the bars signify the value for that time series at the given time on the x-axis.

#### horizontalColorCode example (click to expand)

```
<plot id="
HorizontalColorCodePumps">

<description>testing the
horizontal color code plots<
/description>

    <subplot>

<subPlotType>horizontalColorCode
</subPlotType>

<barMarginPercentage>30<
/barMarginPercentage>

<classBreaks>

<break lowerValue="0" color="
green" label="off"/>

<break lowerValue="1" color="
goldenrod" label="1 pump"/>

<break lowerValue="2" color="
magenta" label="2 pumps"/>

<break lowerValue="3" color="
aquamarine" label="3 pumps"/>

<break lowerValue="4" color="
red"/>

    <
    /classBreaks>

<timeSeriesSet>

<moduleInstanceId>Preprocess_Uni
ts_HTMS</moduleInstanceId>

<valueType>scalar</valueType>

<parameterId>PO_BFN<
/parameterId>

<locationId>BFN Tower 1<
/locationId>

<locationId>BFN Tower 2<
/locationId>

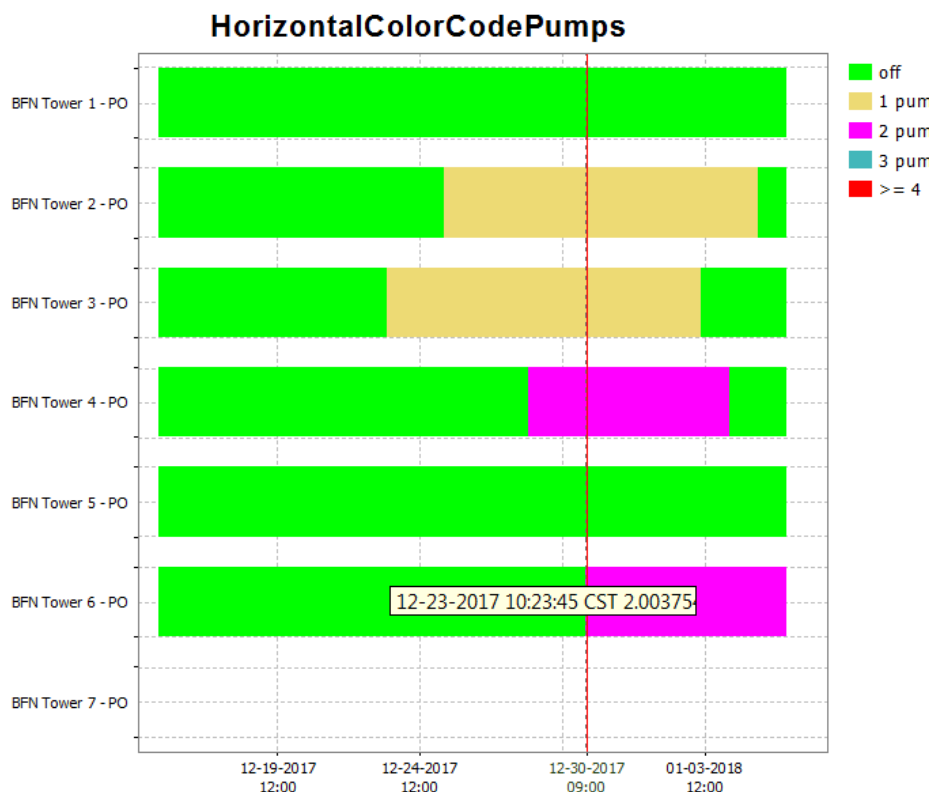
<locationId>BFN Tower 3<
/locationId>

<locationId>BFN Tower 4<
/locationId>

<locationId>BFN Tower 5<
/locationId>

<locationId>BFN Tower 6<
/locationId>

<locationId>BFN Tower 7<
/locationId>
```



```

<timeSeriesType>simulated
forecasting</timeSeriesType>

<timeStep unit="hour"
multiplier="1"/>

<readWriteMode>read only<
/readWriteMode>

    <
  /timeSeriesSet>
    </subplot>
  </plot>

```

## leftAxis and rightAxis

 2022.02 and later

Use elements `leftAxis` and `rightAxis` to configure the axes separately. If the `rightAxis` is configured, the `leftAxis` must be configured too. If the optional attributes `min` and `max` are not configured, then subplot attributes `min` and `max` are used.

An example :

### leftAxis, rightAxis example (click to expand)

```

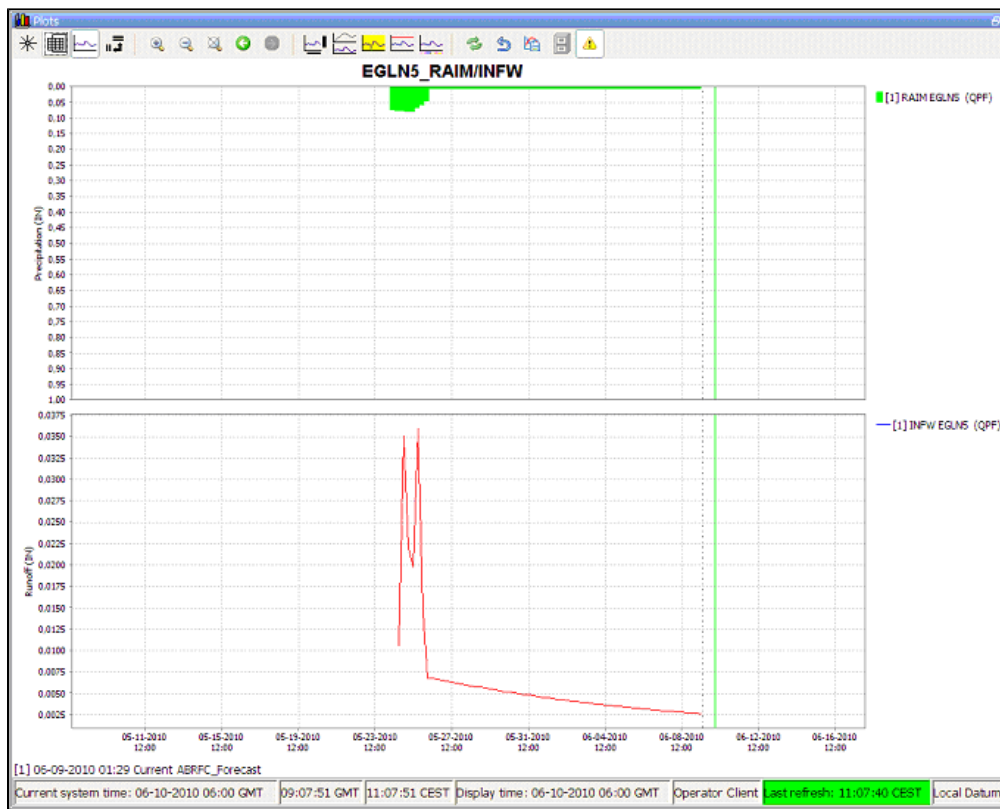
<subplot>
  <leftAxis min="-10" max="100"/>
    <axisLabel>Label left axis</axisLabel>
    <lowerMarginPercentage>10</lowerMarginPercentage>
    <upperMarginPercentage>20</upperMarginPercentage>
  </leftAxis>

  <rightAxis min="20" max="50"/>
    <axisLabel>Label right axis</axisLabel>
    <lowerMarginPercentage>5</lowerMarginPercentage>
    <upperMarginPercentage>5</upperMarginPercentage>
  </rightAxis>
  .....

```

## inverted

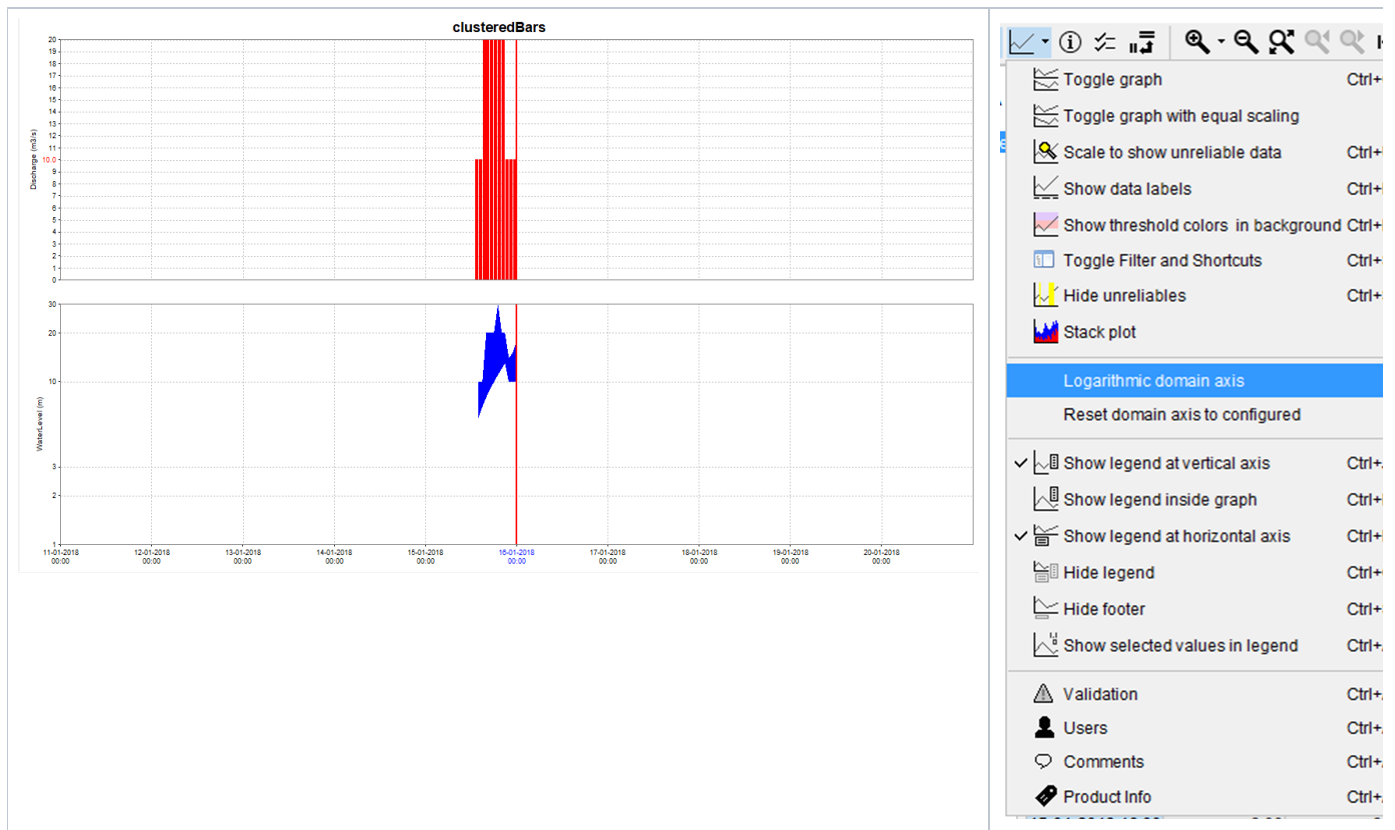
This tag can be used to invert the y-axis of a plot. Below a screenshot of an inverted graph. In the example the timeseries with parameter RAIM is inverted.



## logarithmic

Available since 2020.02. If it is set to true, the subplot y-axis (or domain axis) will be logarithmic. A parameter can also be configured to always/never be displayed on a logarithmic scale, but the subplot setting has precedence. If a parameter is set logarithmic true, but the subplot it appears in is configured logarithmic-false, the axis will not be logarithmic. It is also possible to configure a mixture of logarithmic and non-logarithmic subplots in the same plot.

Logarithmic axis can also be turned on and off manually, using the menu bar. If you do so, the setting will effect all subplots the same way. Should you wish to reset the subplots to the configured version, use the reset domain axis to configured button.

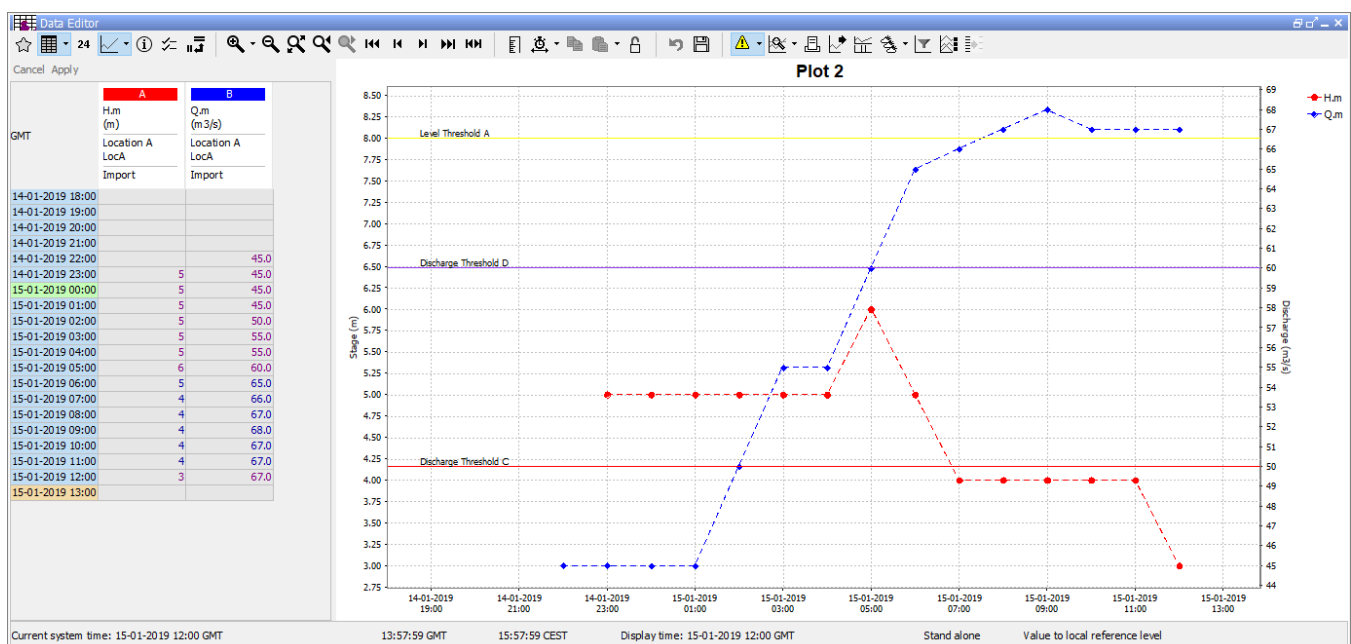


## drawingOrderInverted

This option can be used to invert drawing order of the lines. By default the lines are drawn in the same order as shown in the legend : the first line first, the last line last. The last line is then on top. If drawing order is inverted the last line in the legend is drawn first and the first line last. The first line is then on top.

## thresholdAxis

Thresholds can be displayed on the left axis, or on the right axis, or on both. By default the thresholds are displayed on the left axis. To change it, an option 'thresholdAxis' can be used. Configure `<thresholdAxis>right</thresholdAxis>` if only the right axis should have thresholds, or `<thresholdAxis>both</thresholdAxis>` if both left and right axes should have thresholds





## subplot types

### line

The tag line can be used to configure both single or dual y-axis. Enables the configuration of a custom colour. `visibleInLegend`, `visibleInPlot` and `visibleInTable` are optional, default setting is true. If `visibleInLegend` is set to false, the series will show up in the plot area, but not in the legend. If `visibleInTable` is set to false, the time series can still be seen on the plot, but the data will not show in the table. If `visibleInPlot` is set to false, the time series can be still shown in the table, but will not be visible in the plot. `VisibleInLegend`, `visibleInPlot` and `visibleInTable` are supported by clustered bars too.



#### line remark

Important notice: It is possible to configure `<visibleInTable>`, `<visibleInPlot>` outside of the wrapper elements `<line>`, `<clusteredBars>` or `<area>`, but it should not be done.

There is a bug in one of the external libraries, that causes a mixup, and the wrong `<visibleInTable>`, `<visibleInPlot>` will be paired with the wrong `timeSeriesSet`. Always use `<visibleInTable>` `<visibleInPlot>` together with `<line>`, `<clusteredBars>` or `<area>`.

Example:

#### line example (click to expand)

```
<plot id="Test">
  <subplot>
    <line>
      <color>dark orange</color>
      <visibleInLegend>false</visibleInLegend>
      <visibleInTable>false</visibleInTable>
      <visibleInPlot>true</visibleInPlot>
      <timeSeriesSet>
        <moduleId>Import</moduleId>
        <valueType>scalar</valueType>
        <parameterId>H.m</parameterId>
        <locationId>LocA</locationId>
        <timeSeriesType>external historical</timeSeriesType>
        <timeStep unit="hour"/>
        <relativeViewPeriod unit="hour" start="-12" end="12"/>
        <readWriteMode>editing visible to all future task runs</readWriteMode>
      </timeSeriesSet>
    </line>
  </subplot>
</plot>
```

### area

Attributes available when creating an area type display:

- **color**: Overrides other predefined colours
- **opaquenessPercentage**: Opaueness expressed as a percentage
- **IncludesZeros**: When true the zero is always included in the painted extent

#### filled vertical lines (area)

Use the `<area>` element in combination with a `<lineStyle>` to obtain vertical lines.

### area example (click to expand)

```
<subplot>
  <area>
    <lineStyle>solid<
  /lineStyle>
    <lineWidth>10<
  /lineWidth>
    <color>blue</color>

    <opacityPercentage>50<
  /opacityPercentage>
    <timeSeriesSet>

    <moduleInstanceId>ImportMSW<
  /moduleInstanceId>

    <valueType>scalar<
  /valueType>

    <parameterId>Q.m<
  /parameterId>

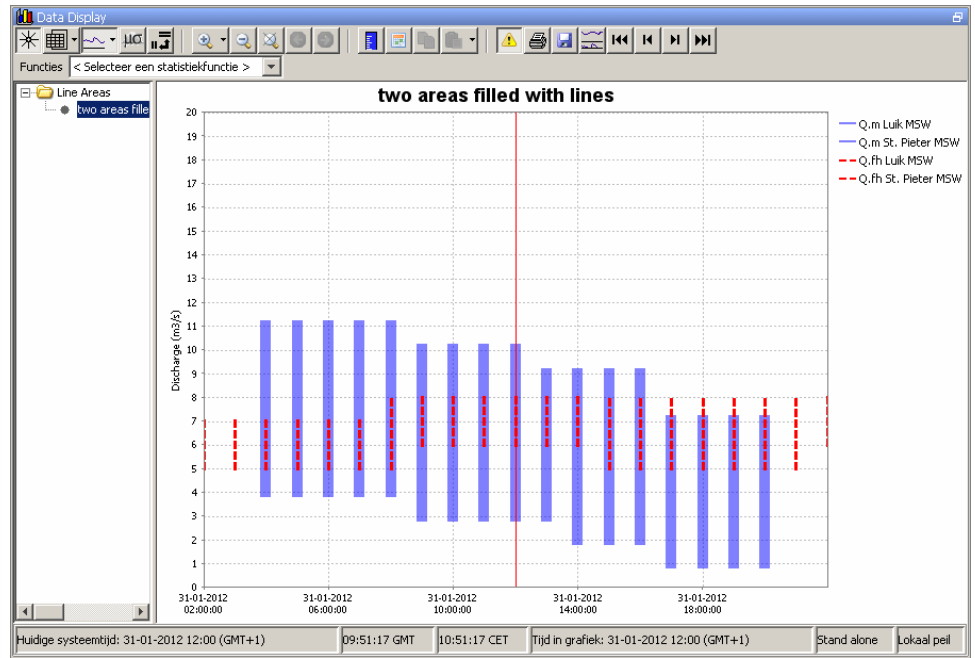
    <locationId>H-MS-LUIK<
  /locationId>

    <locationId>H-MS-SINT<
  /locationId>

    <timeSeriesType>external
  historical</timeSeriesType>
    <timeStep
  unit="hour"/>

    <relativeViewPeriod unit="
  day" start="-2" end="0"/>

    <readWriteMode>read only<
  /readWriteMode>
    </timeSeriesSet>
  </area>
</subplot>
```



## clusteredBars

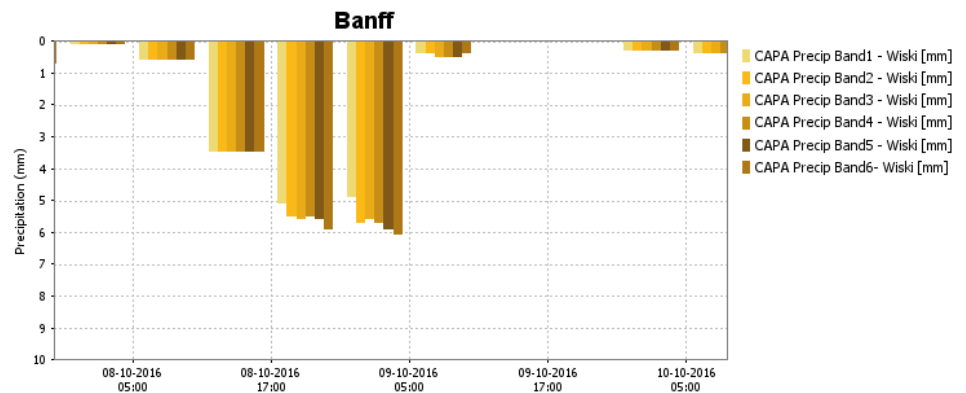
This option makes it possible to cluster bars per timestep. The config example below shows how clustered bars are displayed and configured:

### clusteredBars example (click to expand)

```
<displayGroup name="bars">
  <display name="
  clusteredBars">
    <subplot>
      <clusteredBars
  axis="left">
        <bar>
          <color>blue<
        /color>

        <visibleInLegend>>false<
      /visibleInLegend>

        <visibleInPlot>>false<
      /visibleInPlot>
```



```

<timeSeriesSet>

<moduleInstanceId>Import<
/moduleInstanceId>

<valueType>scalar<
/valueType>

<parameterId>H.m<
/parameterId>

<locationId>LocA<
/locationId>

<timeSeriesType>external
historical</timeSeriesType>
      <timeStep
unit="hour"/>

<relativeViewPeriod unit="
hour" start="-12" end="12"/>

<readWriteMode>editing
visible to all future task
runs</readWriteMode>
      <
/timeSeriesSet>
      </bar>
      <bar>
      <color>red<
/color>

<visibleInTable>false<
/visibleInTable>
      <label>Rated
Discharge</label>

<timeSeriesSet>

<moduleInstanceId>Import<
/moduleInstanceId>

<valueType>scalar<
/valueType>

<parameterId>Q.m<
/parameterId>

<locationId>LocA<
/locationId>

<timeSeriesType>external
historical</timeSeriesType>
      <timeStep
unit="hour"/>

<relativeViewPeriod unit="
hour" start="-12" end="12"/>

<readWriteMode>editing
visible to all future task
runs</readWriteMode>
      <
/timeSeriesSet>
      </bar>
      </clusteredBars>
    </subplot>
  </display>

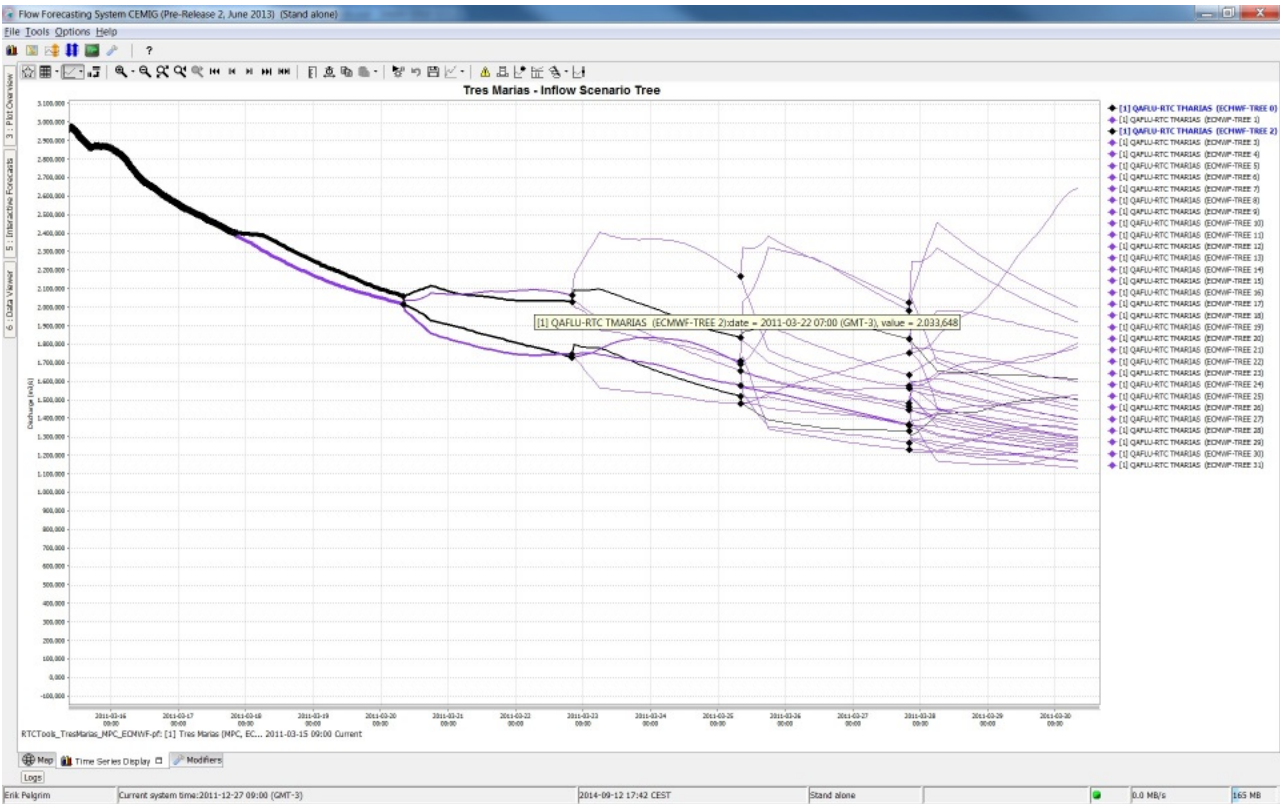
```

## variableLineWidth

Type of plot that can use variable line width's for different graphs. See screenshot below of plot for different ensemble members with different line width's. In this case line width is an indication of the probability of the ensemble member. The sum of probability of all ensemble members at a time is one. Selecting separate ensemble members will mark them with the color black.

Attributes:

- **markerColor**: option to set marker color different from line color
- **lineWidthScaleFactor**: factor to increase or decrease the line width
- **valueTimeSeriesSet**: Time series for XY line plot
- **lineWidthTimeSeriesSet**: Time series for line width of the XY plot lines



## line elements

### axis (dual y-axis plot)

It is possible to have two y-axis visible in the same plot. This can be done by having multiple timeseries where some are assigned to the left y axis and some to the right y axis, or by having a timeseries on the left y axis with an associated rating curve on the right y axis. So every timeseries plot (also in reports) will have a mandatory left y axis to which a timeseries is associated and an optional right y axis to which a timeseries or rating curve is associated.

The display of [threshold](#) lines are supported for one y axis. But there can be threshold lines for either the left or the right y axis, but not both.

### Multiple TimeSeries plots (dual y-axis plot)

To associate a timeseries to the left or right y axis use the <axis> element. This element is optional and leaving it out will associate it (default) to the left axis. Please note that at least one timeseries should be on the left axis (by either settings <axis>left</axis> or leaving the element out and using the default).

Below are two examples of a config and the resulting plot.

**dual axis example (click to expand)**

```
<subplot>
    <plotWeight>1<
/plotWeight>
```

<line>

<color>blue</color>

<axis>left</axis>

<timeSeriesSet>

<moduleInstanceId>ImportTelemetry<  
/moduleInstanceId>

<valueType>scalar</valueType>

<parameterId>H.obs</parameterId>

<locationId>Hydro1</locationId>

<timeSeriesType>external historical<  
/timeSeriesType>

<timeStep unit="minute" multiplier="15"  
/>

<relativeViewPeriod unit="hour" start="  
-36" end="36" />

<readWriteMode>editing visible to all  
future task runs</readWriteMode>

<

/timeSeriesSet>

</line>

<line>

<color>blue</color>

<axis>left</axis>

<timeSeriesSet>

<moduleInstanceId>ImportTelemetry<  
/moduleInstanceId>

<valueType>scalar</valueType>

<parameterId>H.rated</parameterId>

<locationId>Hydro3</locationId>

<timeSeriesType>external historical<  
/timeSeriesType>

<timeStep unit="minute" multiplier="15"  
/>

<relativeViewPeriod unit="hour" start="  
-36" end="36" />

<readWriteMode>editing visible to all  
future task runs</readWriteMode>

<

/timeSeriesSet>

</line>

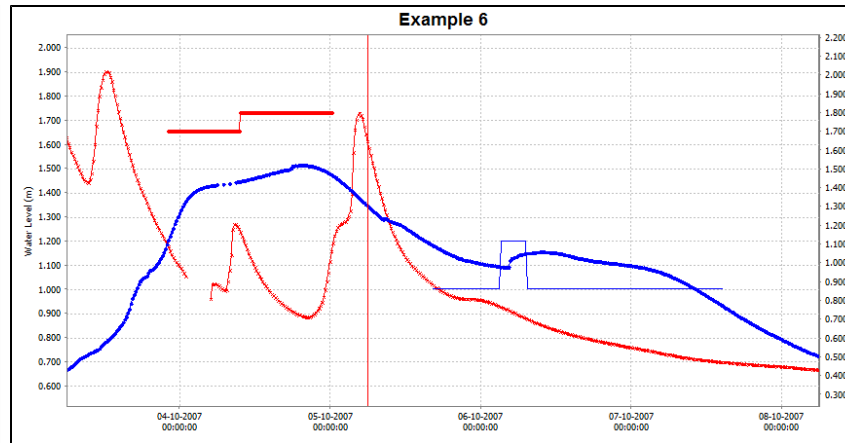
<line>

<color>red</color>

<axis>right</axis>

<timeSeriesSet>

<moduleInstanceId>ImportTelemetry<



```

/moduleInstanceId>

<valueType>scalar</valueType>

<parameterId>Q.obs</parameterId>

<locationId>Hydro2</locationId>

<timeSeriesType>external historical<
/timeSeriesType>

<timeStep unit="minute" multiplier="15"
/>

<relativeViewPeriod unit="hour" start="
-36" end="36"/>

<readWriteMode>editing visible to all
future task runs</readWriteMode>
      <
/timeSeriesSet>
      </line>
      <line>

<color>red</color>

<axis>right</axis>

<timeSeriesSet>

<moduleInstanceId>ImportTelemetry<
/moduleInstanceId>

<valueType>scalar</valueType>

<parameterId>Q.rated</parameterId>

<locationId>Hydro4</locationId>

<timeSeriesType>external historical<
/timeSeriesType>

<timeStep unit="minute" multiplier="15"
/>

<relativeViewPeriod unit="hour" start="
-36" end="36"/>

<readWriteMode>editing visible to all
future task runs</readWriteMode>
      <
/timeSeriesSet>
      </line>
    </subplot>

```

**dual axis with 2 parameterGroups example (click to expand)**

```

        <subplot>
          <plotWeight>1<
/plotWeight>
          <line>

<color>blue</color>

<axis>right</axis>

<timeSeriesSet>

<moduleInstanceId>ImportTelemetry<
/moduleInstanceId>

<valueType>scalar</valueType>

<parameterId>H.obs</parameterId>

<locationId>Hydro1</locationId>

<timeSeriesType>external historical<
/timeSeriesType>

<timeStep unit="minute" multiplier="15"
/>

<relativeViewPeriod unit="hour" start="
-36" end="36"/>

<readWriteMode>editing visible to all
future task runs</readWriteMode>
      <
/ timeSeriesSet>
          </line>
          <line>

<color>red</color>

<timeSeriesSet>

<moduleInstanceId>ImportTelemetry<
/moduleInstanceId>

<valueType>scalar</valueType>

<parameterId>Q.obs</parameterId>

<locationId>Hydro2</locationId>

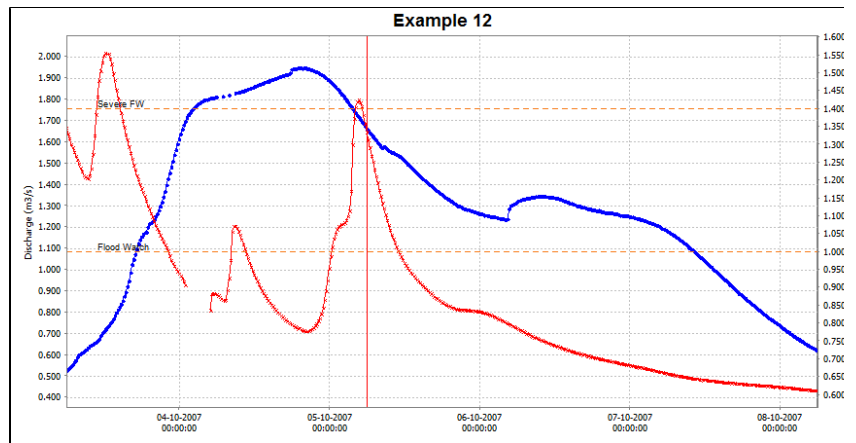
<timeSeriesType>external historical<
/timeSeriesType>

<timeStep unit="minute" multiplier="15"
/>

<relativeViewPeriod unit="hour" start="
-36" end="36"/>

<readWriteMode>editing visible to all
future task runs</readWriteMode>
      <
/ timeSeriesSet>
          </line>
        </subplot>

```



Threshold line only for timeseries on the right y axis

Stage/discharge plots (dual y-axis plot)

When a discharge is displayed, it is possible to show the stage on the right axis. It is not possible to configure this rating on the left axis (and the timeseries on the right axis), since a plot has to have at least one timeseries on the left axis.

The right axis is then not a linear axis but the ticks on the right axis are calculated from the discharge ticks on the left axis.

It is also possible to display the stage and show the discharge on the right axis. The example below shows a display which plots several discharge time series.

The left axis is a linear axis with ticks for the discharge. The right axis is a non-linear axis.

The ticks on the right axis are calculated from the value of the discharge on the left axis by using a rating curve.

Below an configuration and resulting plot

ratingAxis example (click to expand)

```
<subplot>
                                <line>

<color>black</color>

<lineStyle>none</lineStyle>

<markerStyle>+</markerStyle>

<markerSize>3</markerSize>

<timeSeriesSet>

<moduleInstanceId>ImportTelemetry<
/moduleInstanceId>

<valueType>scalar</valueType>

<parameterId>Q.rated</parameterId>

<locationId>Hydro4</locationId>

<timeSeriesType>external historical<
/timeSeriesType>

<timeStep unit="minute" multiplier="15"
/>

<relativeViewPeriod unit="hour" start="
-36" end="36"/>

<readWriteMode>editing visible to all
future task runs</readWriteMode>
<
/timeSeriesSet>
                                </line>
                                <line>

<color>black</color>

<lineStyle>solid</lineStyle>

<markerStyle>circle</markerStyle>

<markerSize>6</markerSize>

<axis>left</axis>

<ratingAxis>

<parameterGroupId>Water Level<
/parameterGroupId>
```

Example 3



```

<transformationType>dischargeStage<
/transformationType>
      <
/ratingAxis>

<timeSeriesSet>

<moduleInstanceId>ImportTelemetry<
/moduleInstanceId>

<valueType>scalar</valueType>

<parameterId>Q.obs</parameterId>

<locationId>Hydro2</locationId>

<timeSeriesType>external historical<
/timeSeriesType>

<timeStep unit="minute" multiplier="15"
/>

<relativeViewPeriod unit="hour" start="
-36" end="36"/>

<readWriteMode>editing visible to all
future task runs</readWriteMode>
      <
/timeSeriesSet>
      </line>
    </subplot>

```

## timeSeriesSet

### chainageLocationSetId

(Since 2016.02) By using a chainage location set in a time series set of a plot in the display groups, a longitudinal profile can be plot based on scalar time series for different locations.

Notice that it is not possible to combine a longitudinal plot in with a standard plot (having time on the X-axis). Furthermore, it is not possible to use it for relatedLocations, and also thresholds will not be shown.

### chainageLocationSetId example (click to expand)

```
<plot id="
Langsprofiel_Caspargowse_Weteri
ng">

    <subplot>

<timeSeriesSet>

<moduleInstanceId>WerkFilter<
/moduleInstanceId>

<valueType>scalar</valueType>

<parameterId>H.G.15<
/parameterId>

<chainageLocationSetId>Langsprof
iel_Caspargowse_Wetering<
/chainageLocationSetId>

<timeSeriesType>external
historical</timeSeriesType>

<timeStep unit="minute"
multiplier="15"/>

<relativeViewPeriod unit="day"
start="-32" end="0"/>

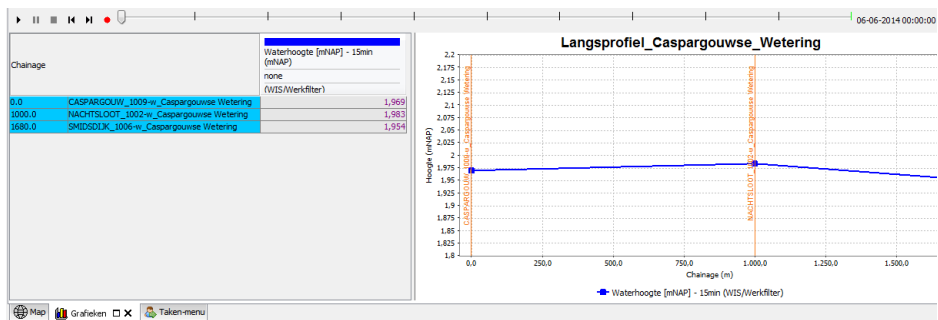
<readWriteMode>read only<
/readWriteMode>

    <
/chainageLocationSetId>
/chainageLocationSetId>

    </timeSeriesSet>
    </subplot>
    </plot>
    <displayGroup name="
LangsProfiel">
        <display name="
Langsprofiel_Caspargowse_Weteri
ng">

<locationSetId>Langsprofiel_Casp
argowse_Wetering<
/chainageLocationSetId>

<plotId>Langsprofiel_Caspargows
e_Wetering</plotId>
        </display>
    </displayGroup>
```



## domainParameterId

### Single Domain Axis Time Series / 1D Spectra plots

Time series with one domain axis are displayed as one xy chart per time step, like longitudinal profiles. A slider is used to navigate through time.

**domainParameterId example (click to expand)**

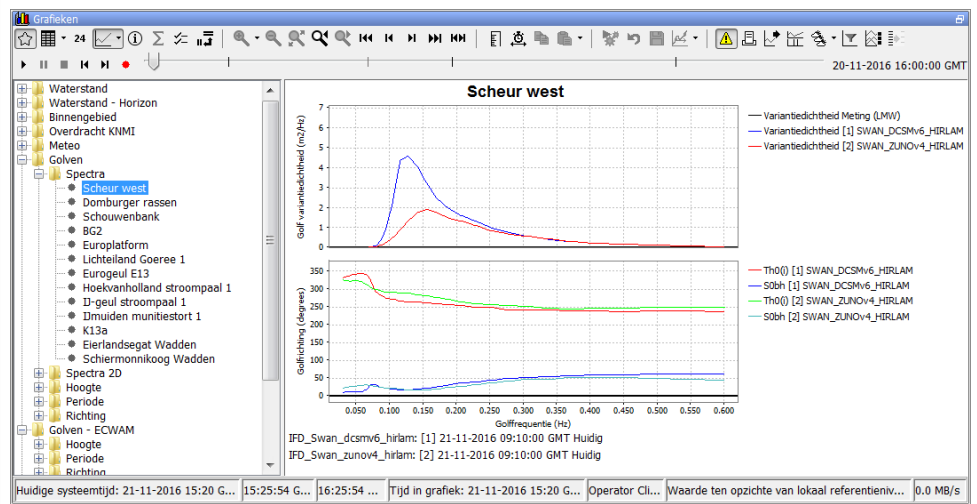
```
<timeSeriesSet>

<moduleInstanceId>LMW_Import
_spectrum</moduleInstanceId>
  <valueType>scalar<
/ valueType>
  <parameterId>Wave.
variancedensity.meting<
/ parameterId>

<domainParameterId>f<
/ domainParameterId>

<locationSetId>LMW_Czz10<
/ locationSetId>

<timeSeriesType>external
historical</timeSeriesType>
  <timeStep unit="
minute" multiplier="60"/>
  <relativeViewPeriod
unit="day" start="-2" end="
2"/>
  <readWriteMode>read
only</readWriteMode>
</timeSeriesSet>
```

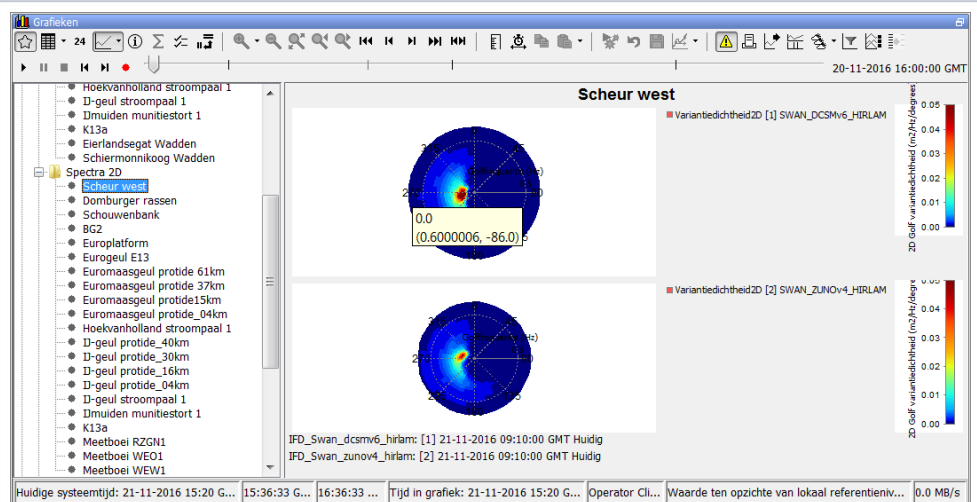


## Dual Domain Axis Time Series / 2D Polar Spectra plots

When a second domain axis is available this will always be used as direction for a polar plot by the time series dialog. When second domain axis is not a directional parameter or the time series has more than two domain axes the time series can not be displayed in the time series dialog with the current functionality

**domainParameterId example (click to expand)**

```
<subplot>
  <classBreaks>
    <break
symbolSize="10" color="
000080" lowerValue="0"/>
    <break
symbolSize="10" color="
0000A8" lowerValue="0.0005"
/>
    <break
symbolSize="10" color="
0000E6" lowerValue="0.001"/>
    <break
symbolSize="10" color="
004DFF" lowerValue="0.0025"
/>
    <break
symbolSize="10" color="
00B3FF" lowerValue="0.005"/>
    <break
symbolSize="10" color="
1AFFE5" lowerValue="0.01"/>
    <break
symbolSize="10" color="
80FF80" lowerValue="0.015"/>
  </classBreaks>
```



```

symbolSize="10" color="
E6FF19" lowerValue="0.02"/>
    <break
symbolSize="10" color="
FFB200" lowerValue="0.025"/>
    <break
symbolSize="10" color="
FF4C00" lowerValue="0.035"/>
    <break
symbolSize="10" color="
E50000" lowerValue="0.04"/>
    <break
symbolSize="10" color="
A80000" lowerValue="0.045"/>
    <break
symbolSize="10" color="
800000" lowerValue="0.05"/>
    </classBreaks>
    <direction>from<
/direction>
    <timeSeriesSet>

<moduleInstanceId>Swan_dcsmv
6_hirlam_forecast_import<
/moduleInstanceId>

<valueType>scalar<
/valueType>

<parameterId>Wave.
variancedensity2D.voorspeld<
/parameterId>

<domainParameterId>f<
/domainParameterId>

<domainParameterId>dir<
/domainParameterId>

<locationSetId>SWAN_POINTS2<
/locationSetId>

<timeSeriesType>simulated
forecasting</timeSeriesType>
    <timeStep
unit="minute" multiplier="
60"/>

<relativeViewPeriod unit="
day" start="-2" end="2"/>

<readWriteMode>read only<
/readWriteMode>
    <
/timeSeriesSet>
</subplot>

```

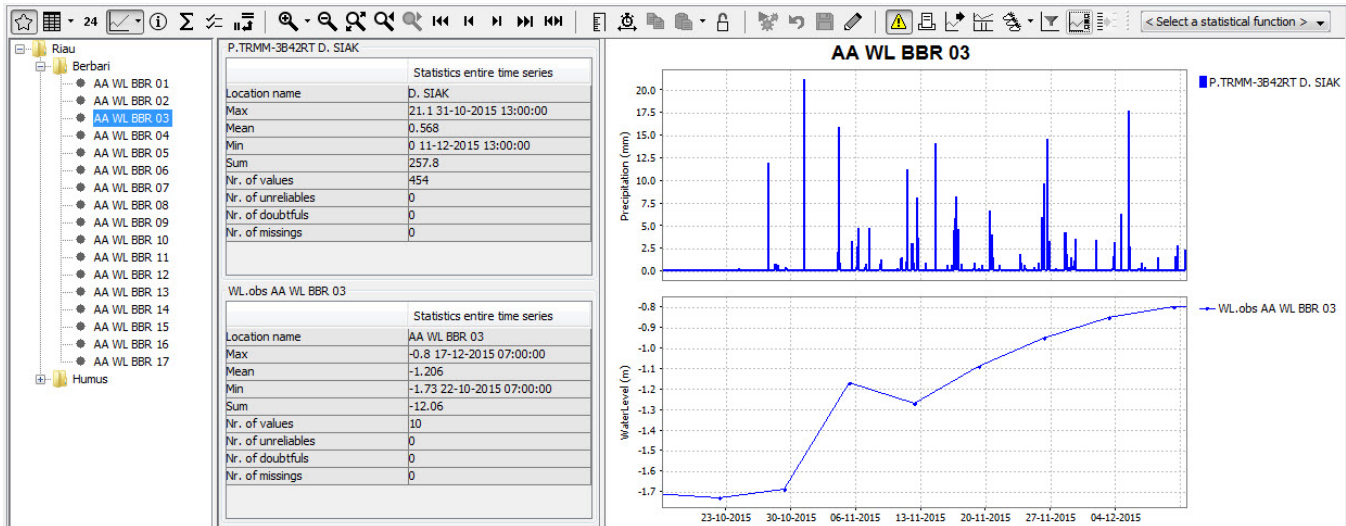
## visibleInTable, visibleInPlot



Important notice: It is possible to configure `<visibleInTable>`, `<visibleInPlot>` outside of the wrapper elements `<line>`, `<clusteredBars>` or `<area>`, but it should not be done. There is a bug in one of the external libraries, that causes a mixup, and the wrong `<visibleInTable>`, `<visibleInPlot>` will be paired with the wrong timeSeriesSet. Always use `<visibleInTable>` `<visibleInPlot>` together with `<line>`, `<clusteredBars>` or `<area>`.

## descriptiveFunctions

Adding descriptiveFunctions will result in an extra panel in the display, showing relevant descriptive information about the timeseries in view. More information on the descriptive functions available can be found at [TimeSeriesDisplayConfiguration](#)



## additionalTimeSeries

Additional time series can be configured in display group plots. These series are found via attribute functions in the `<locationFunctionEquals selected=@ATTRIBUTE_ID_A@ additional="@ATTRIBUTE_ID_B@">` of the selected time series and the additional time series. When the attribute functions evaluate to the same value the additional time series are added as invisible by default. When additional time series are available the icon of the visibility dialog will change which shows that invisible time series are available and can be made visible.

Since 2020.02, it is possible to match the time series based on any view period (not just the current system time) by setting the [showLocationsOutsideVisibilityPeriod](#) option to true.

When it is desired that the additional time series are added only for the same locations then all locations should have an attribute that is equal to their id for example LOC\_ID and the `locationFunctionEquals` element can be used as follows:

```
<locationFunctionEquals selected=@LOC_ID@ additional="@LOC_ID@">
```

In the example below `locationFunctionEquals selected="@LOC_ID@" additional="@AFGB_NAAR@">` is used which will show all a timeseries from the additional time series set where the location has an attribute value AFGB\_NAAR which equals the LOC\_ID attribute of the shown location of the normal time series set, which can be multiple or even 0.

There is an extra option to show the additional time series in a separate sub plot to make sure there are no plots with incompatible parameter combinations.

Each additional time series element can be given a name, this name will be used as prefix in the legend of the time series and as GroupNode label in the visibility dialog.



### Making stacked graphs

Deprecated subplot attribute `stackPlot` (deprecated since 2014.02) - was used to create an area plot into a plot of stacked areas. Instead use `<subPlotType>stack</subPlotType>`

### additionalTimeSeries example (click to expand)

```
<plot id="WATERBALANS_DAG_CAW">
  <subplot stackPlot="
true">
    <lineStyle>bar<
  /lineStyle>
    <timeSeriesSet>
      <moduleInstanceId>BalansAfvoergebieden<
      /moduleInstanceId>
      <valueType>scalar</valueType>
      <parameterId>H.G.i.d</parameterId>
```

```

<locationSetId>AFVOERGEBIEDEN<
/locationSetId>

<timeSeriesType>external historical<
/timeSeriesType>

<timeStep unit="day" timeZone="GMT+1"/>

<relativeViewPeriod unit="day" start="
-32" end="0"/>

<readWriteMode>read only</readWriteMode>
</timeSeriesSet>

<additionalTimeSeries name="
*AFGB_NAAR*" separateSubPlot="true">

<locationFunctionEquals selected="
@LOC_ID@" additional="@AFGB_NAAR@"/>

<timeSeriesSet>

<moduleInstanceId>ImportOpvlWater<
/moduleInstanceId>

<valueType>scalar</valueType>

<parameterId>Q.G.15</parameterId>

<locationSetId>OPVLWATER_SUBLOC_DEBIETEN
</locationSetId>

<timeSeriesType>external historical<
/timeSeriesType>

<timeStep unit="minute" multiplier="15"
/>

<relativeViewPeriod unit="hour" start="
-70" end="0"/>

<readWriteMode>read only</readWriteMode>
<
/timeSeriesSet>
<
/additionalTimeSeries>

<additionalTimeSeries name="*AFGB_VAN*"
separateSubPlot="true">

<locationFunctionEquals selected="
@LOC_ID@" additional="@AFGB_VAN@"/>

<timeSeriesSet>

<moduleInstanceId>ImportOpvlWater<
/moduleInstanceId>

<valueType>scalar</valueType>

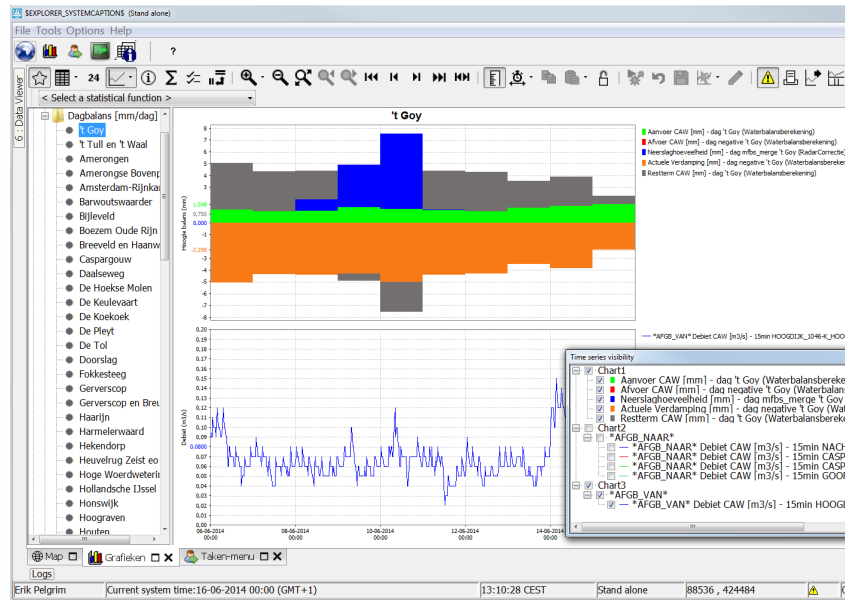
<parameterId>Q.G.15</parameterId>

<locationSetId>OPVLWATER_SUBLOC_DEBIETEN
</locationSetId>

<timeSeriesType>external historical<
/timeSeriesType>

<timeStep unit="minute" multiplier="15"
/>

```



```

<relativeViewPeriod unit="hour" start="
-70" end="0"/>

<readWriteMode>read only</readWriteMode>
    <
/timeSeriesSet>
    <
/additionalTimeSeries>
    </subplot>
</plot>

```

## locationLoop (displayGroup)

By defining a location loop within a displayGroup, the display group will be created for each location of location set configured in the loop.

- **locationSetId**: id of the location set for which the loop should be created
- **includeChildren**: boolean to define whether also the child locations of the locations from the location set should be included
- **locationRelationId**: optional and can be multiple. These elements determine whether also related locations should be included. This applies to relations both to and from the location from the location set.

### locationLoop example (click to expand)

```

<displayGroup name="Validations Location Loop">
  <displayGroup name=" ">
    <locationLoop>
      <locationSetId>FloodingAreas</locationSetId>
      <includeChildren>true</includeChildren>
      <locationRelationId>L_AREA</locationRelationId>
      <locationRelationId>L_VISIBILIY</locationRelationId>
      <locationRelationId>MEASUREMENT_PARENT</locationRelationId>
    </locationLoop>
    <displayGroup name="Waterbalance">
      <display name="Waterbalance">
        <plotId>Waterbalance</plotId>
      </display>
      <display name="Waterbalance incl. error">
        <plotId>Waterbalance_error</plotId>
      </display>
    </displayGroup>
    ...
  </displayGroup>
</displayGroup>

```

## display (displayGroup)

Definition of a pre-configured display. Each display may contain multiple sub-plots. Multiple displays may be defined per display group.

Some of the elements of display:

- **parentLocationId**:
- **description**
- **startForwardLookingPeriod**: way to define view period for seasonal or water year analysis
- **nrOfRecentForecasts**: Can be applied to show multiple recent forecasts (simulated forecast type)
- **plotId**
- **tickTimeStep**
- **thresholdLocationSetId**

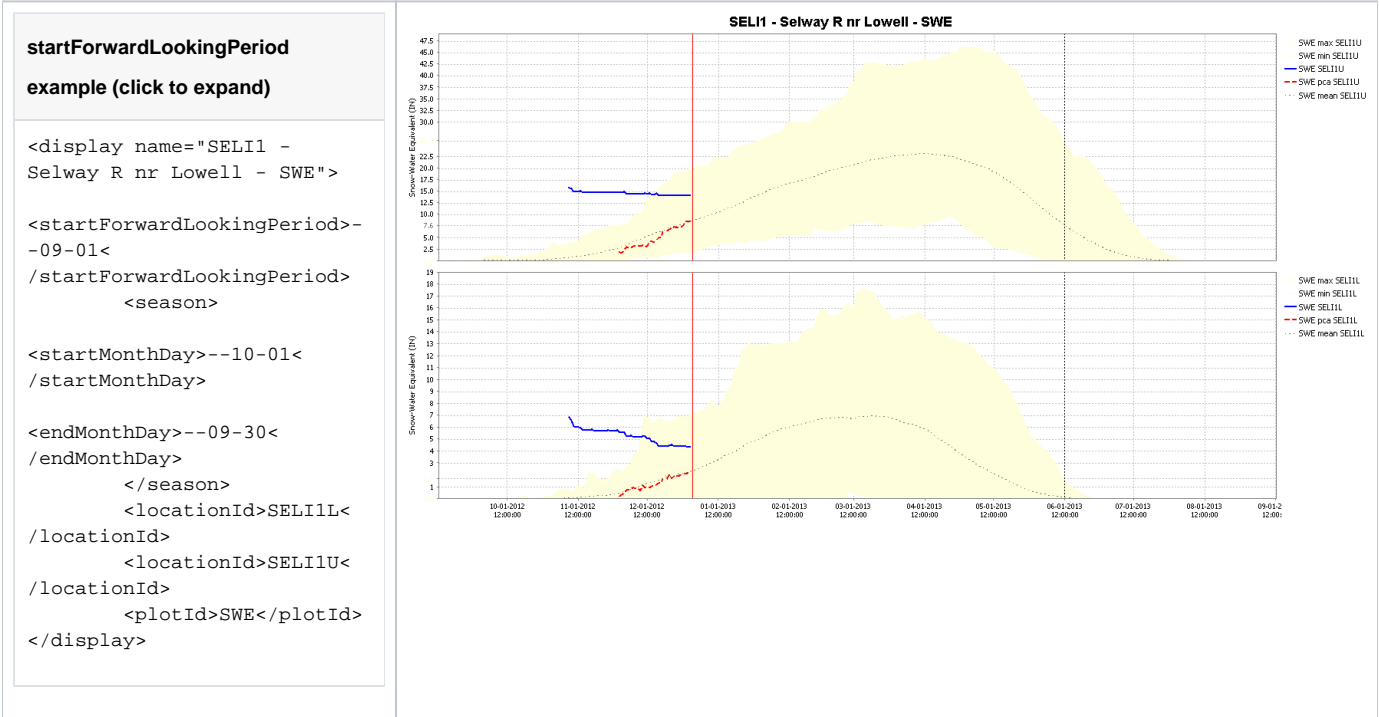
### startForwardLookingPeriod

Normally, displays are defined with a relative view period, so that the view window progresses with current system time. This paradigm does not fit systems that are more oriented towards seasonal or water year analysis. Therefore an option has been added since 2011.02 to define a season, such that the time axis is fixed, while the vertical red line (current system time) moves over time from left to right through the display. To utilize this functionality, a season should be defined instead of a relative view period. This season specification is composed of a start and end date, as well as an entry called 'startForwardLookingPeriod'.

The software compares the currentSystemTime against the date specified in the 'startForwardLookingPeriod' entry, to decide if the time axis needs to shift to the next year/season. This decision is relevant within a water supply forecasting context since a few weeks before the end of the water year you may like to look forward to the next wateryear instead of the current wateryear. Use readWriteMode=read only in your time series sets, as other modes such as 'read complete forecast' stretches the time horizon, disrupting the time horizon displayed outside the season.

E.g. in the code example below, a wateryear is defined from 1st of October to 30th of September. The startForwardLookingPeriod is defined as 1st of September.

currentSystemTime	view period x-axis (start-end)
12 Aug.2012	1 Oct.2011 - 30 Sept.2012
12 Sept.2012	1 Oct.2012 - 30 Sept.2013
12 Oct.2012	1 Oct.2012 - 30 Sept.2013



## tickTimeStep

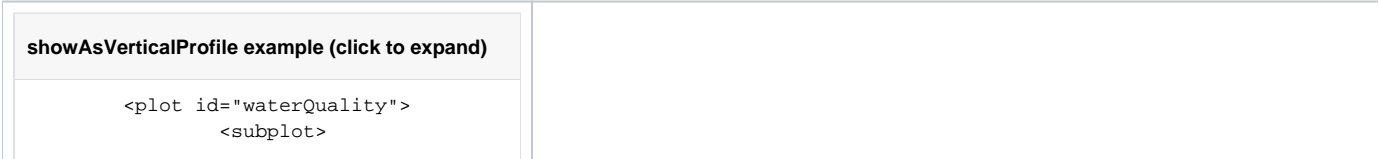
To tie the date/time ticks to the valid times of the (cardinal) time step, configure 'tickTimeStep' in DisplayGroups.xml. For example, if the 'tickTimeStep' is 6 hours then the ticks are always aligned with the synoptic times 00Z, 06Z, 12Z, 18Z. When we are zooming out and there is no space to display all date /times along the x-axis, then one or more synoptic times ticks are omitted. For example we see ticks at 00Z and 18Z.

## singleLocationDisplays (displayGroup)

Adds multiple displays at once to this display group. Every display will show only one location.

## showAsVerticalProfile

When true vertical profile timeseries are calculated on the fly from the scalar time series at the child locations of the locations defined in the timeSeriesSet.





<line>

<color>blue</color>

<lineStyle>solid;thick</lineStyle>

<markerStyle>square</markerStyle>

<timeSeriesSet>

<moduleInstanceId>ProcessObservedWaterquality</moduleInstanceId>

<valueType>scalar</valueType>

<parameterId>WT\_obs</parameterId>

<locationId>R\_401027</locationId>

<timeSeriesType>external historical</timeSeriesType>

<timeStep unit="hour" multiplier="3"/>

<readWriteMode>read only</readWriteMode>

</timeSeriesSet>

</line>

</subplot>

<subplot>

<line>

<color>turquoise</color>

<lineStyle>solid;thick</lineStyle>

<markerStyle>square</markerStyle>

<timeSeriesSet>

<moduleInstanceId>ProcessObservedWaterquality</moduleInstanceId>

<valueType>scalar</valueType>

<parameterId>EC\_obs</parameterId>

<locationId>R\_401027</locationId>

<timeSeriesType>external historical</timeSeriesType>

<timeStep unit="hour" multiplier="3"/>

<readWriteMode>read only</readWriteMode>

</timeSeriesSet>

</line>

</subplot>

<subplot>

<line>

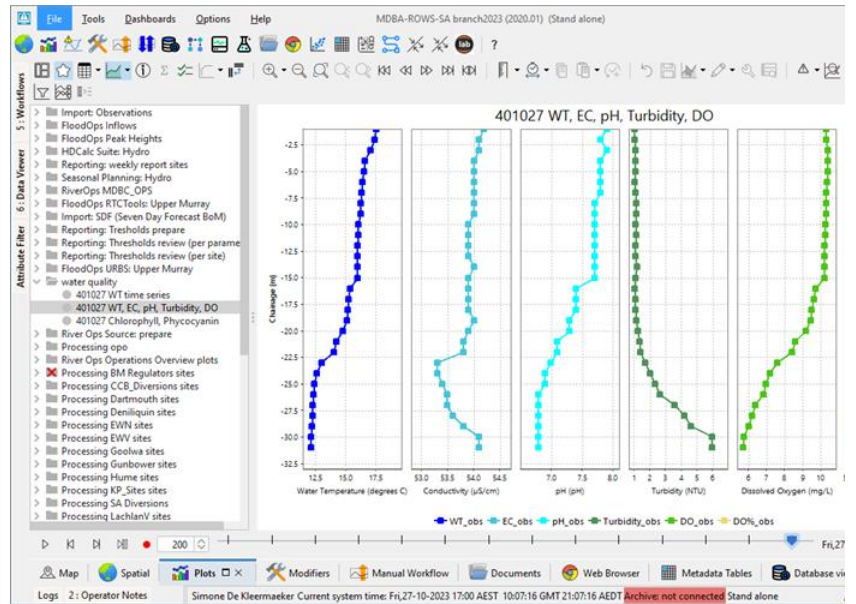
<color>cyan</color>

<lineStyle>solid;thick</lineStyle>

<markerStyle>square</markerStyle>

<timeSeriesSet>

<moduleInstanceId>ProcessObservedWaterqu



```

ality</moduleInstanceId>

<valueType>scalar</valueType>

<parameterId>pH_obs</parameterId>

<locationId>R_401027</locationId>

<timeSeriesType>external historical<
/timeSeriesType>

<timeStep unit="hour" multiplier="3"/>

<readWriteMode>read only</readWriteMode>
<
/timeSeriesSet>
    </line>
</subplot>
<subplot>
    <line>

<color>forest green</color>

<lineStyle>solid;thick</lineStyle>

<markerStyle>square</markerStyle>

<timeSeriesSet>

<moduleInstanceId>ProcessObservedWaterqu
ality</moduleInstanceId>

<valueType>scalar</valueType>

<parameterId>Turbidity_obs</parameterId>

<locationId>R_401027</locationId>

<timeSeriesType>external historical<
/timeSeriesType>

<timeStep unit="hour" multiplier="3"/>

<readWriteMode>read only</readWriteMode>
<
/timeSeriesSet>
    </line>
</subplot>
<subplot>
    <line>

<color>green3</color>

<lineStyle>solid;thick</lineStyle>

<markerStyle>square</markerStyle>

<timeSeriesSet>

<moduleInstanceId>ProcessObservedWaterqu
ality</moduleInstanceId>

<valueType>scalar</valueType>

<parameterId>DO_obs</parameterId>

<locationId>R_401027</locationId>

<timeSeriesType>external historical<
/timeSeriesType>

```

```

<timeStep unit="hour" multiplier="3"/>

<readWriteMode>read only</readWriteMode>
    <
/timeSeriesSet>
        </line>
    </subplot>
    <subplot>
        <line>

<color>goldenrod</color>

<lineStyle>solid;thick</lineStyle>

<markerStyle>square</markerStyle>

<timeSeriesSet>

<moduleInstanceId>ProcessObservedWaterqu
ality</moduleInstanceId>

<valueType>scalar</valueType>

<parameterId>DO%_obs</parameterId>

<locationId>R_401027</locationId>

<timeSeriesType>external historical<
/timeSeriesType>

<timeStep unit="hour" multiplier="3"/>

<readWriteMode>read only</readWriteMode>
    <
/timeSeriesSet>
        </line>
    </subplot>
    </plot>
    <displayGroup id="waterquality"
name="water quality">
        <display id="
WQ_R_401027" name="401027 WT, EC, pH,
Turbidity, DO">

<relativeViewPeriod unit="hour" start="
-336" end="168"/>

<locationId>R_401027</locationId>

<plotId>waterQuality</plotId>

<showAsVerticalProfile>true<
/showAsVerticalProfile>
        </display>
    </displayGroup>

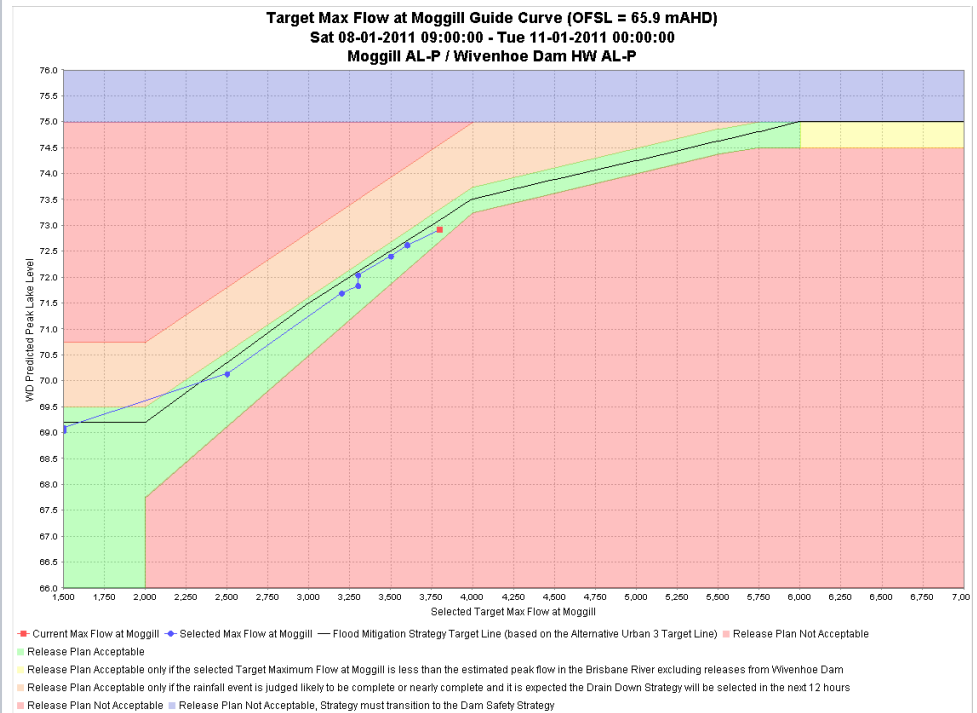
```

showAsScatterPlot

To plot time series, with the values of the different time series on the X and Y axis.

- <referencePoints> can be defined in several ways:
  - <point> will be shown as an interaction line.
  - <xAttributeld>, <yAttributeld> defining the x and y coordinates of the points, the y-coordinates.
  - Time dependent location attributes are not supported.
- <interactionLine>
  - Any number of interaction lines can be added to the plot.
  - Lines configured with points or location attributes can be added in any order or combination.
  - You can customise the interaction lines. In this case, the <referencePoints> element should be surrounded by <interactionLine> element.
  - None of the customisation options within an <interactionLine> are compulsory.
  - With multiple interaction lines, all should be configured as either interactionLine or referencePoints. You cannot mix <referencePoints> and <interactionLine>

Example from operational system



Config examples:

### referencePoints example (click to expand)

```
<referencePoints>
  <xAttributeId>BEP_80_Q<
/xAttributeId>
  <yAttributeId>BEP_80_H<
/yAttributeId>
</referencePoints>
<referencePoints>
  <point>
    <x>0</x>
    <y>60</y>
  </point>
  <point>
    <x>500</x>
    <y>60</y>
  </point>
  <point>
    <x>1000</x>
    <y>70</y>
  </point>
  <point>
    <x>1500</x>
    <y>75</y>
  </point>
</referencePoints>
<referencePoints>
  <xAttributeId>Q_50Hz<
/xAttributeId>
  <yAttributeId>H_50Hz<
/yAttributeId>
</referencePoints>
```

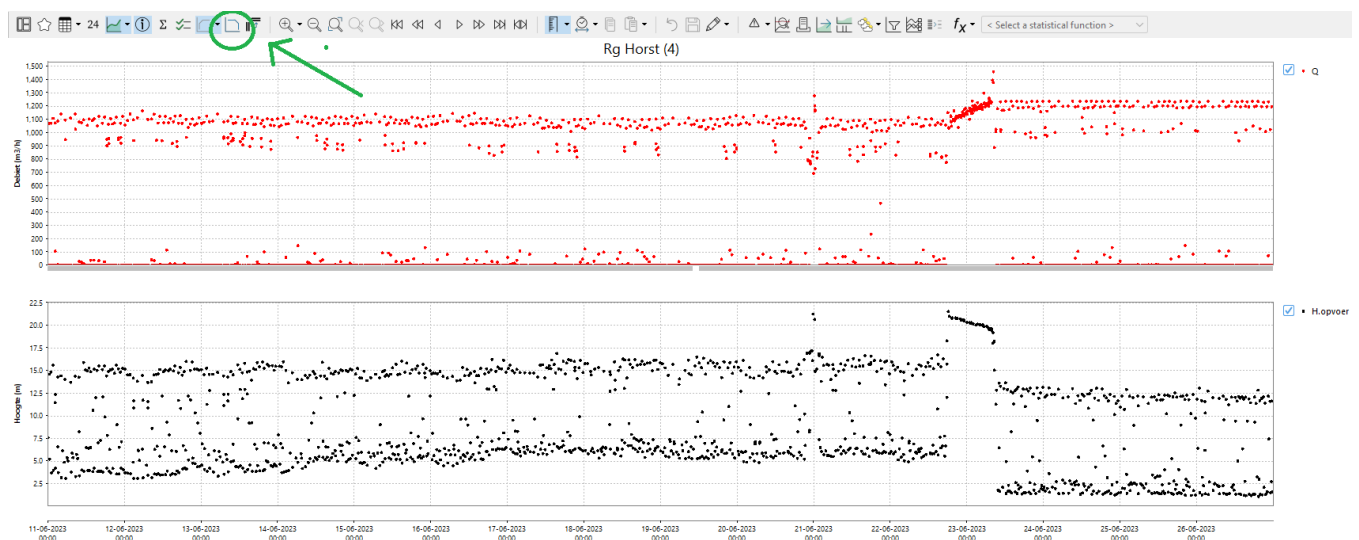
### interactionLine example (click to expand)

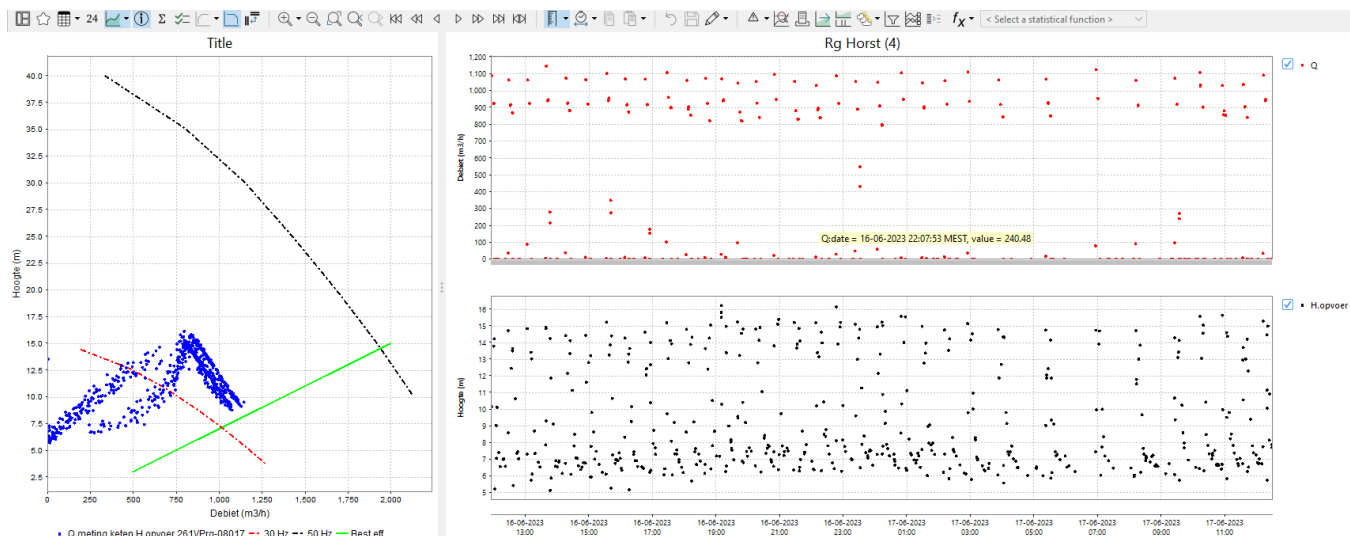
```
<interactionLine>
  <lineColor>red</lineColor>
  <lineStyle>solid;thick<
/lineStyle>
  <lineWidth>2</lineWidth>
  <labelText>30 Hz</labelText>
  <referencePoints>
    <xAttributeId>Q_30Hz<
    <yAttributeId>H_30Hz<
  </referencePoints>
</interactionLine>
<interactionLine>
  <lineColor>blue</lineColor>
  <lineStyle>solid</lineStyle>
  <lineWidth>1</lineWidth>
  <labelText>samenloop<
/labelText>
  <point>
    <x>600</x>
    <y>5</y>
  </point>
  <point>
    <x>700</x>
    <y>4</y>
  </point>
  <point>
    <x>1200</x>
    <y>8</y>
  </point>
</referencePoints>
</interactionLine>
```

To display the data in a scatterplot, you can chose between two options:

- <showAsScatterPlot>: TimeSeries display will be replaced with a scatterplot
- <addInteractionScatterPlot>: you can turn on an extra scatterplot that is displayed next to the original time series.
  - This button has to be enabled in the TimeSeriesDisplayConfig.xml buttonSettings with <showInteractionScatterPlot visible="true"/>

With the elements <xAxisRange> and <yAxisRange> the range of the x and y axis can be defined.





## Pairing of time series

By default all time series with the same location are paired up, but this behavior can be changed with the following configuration options.

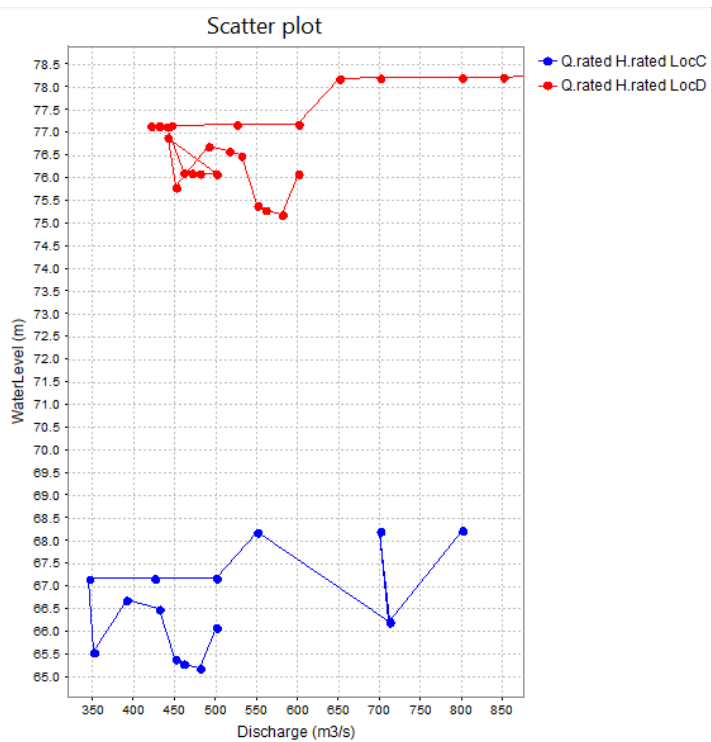
- when there are 2 time series they will be paired up, regardless of their location or parameter (since 2019.02)
- createPairsByMatchingParameter: time series with the same parameter can be paired (since 2017.02)
- markerVisibility: choose whether all data points, none data points or onlyLast data point is visualised (since 2019.02)
- matchTimeSeriesPair: pair up the time series based on their configured order: timeSeriesOrder (since 2020.02)
- draw colored areas based on configured points. Also, in the plot, the grid lines in the background and the interaction line are visible, and legends are added next to the plot. (since 2020.02)

Scatter series are shown in standard time series table and all configuration elements are optional, so it suffices to configure <showAsScatterPlot/> (since 2022.02).

A simple scatter plot without any additional options is shown below :

GMT	A	B	C	D
	Q.rated (m3/s)	H.rated (m)	Q.rated (m3/s)	H.rated (m)
	Location C LocC	Location C LocC	Location D LocD	Location D LocD
	[1] 13-01-2020 00:00	[1] 13-01-2020 00:00	[1] 13-01-2020 00:00	[1] 13-01-2020 00:00
14-01-2020 01:00	500.100	66.10	600.100	76.10
14-01-2020 02:00	480.200	65.20	580.200	75.20
14-01-2020 03:00	460.300	65.30	560.300	75.30
14-01-2020 04:00	450.400	65.40	550.400	75.40
14-01-2020 05:00	430.500	66.50	530.500	76.50
14-01-2020 06:00			515.600	76.60
14-01-2020 07:00	390.700	66.70	490.700	76.70
14-01-2020 08:00	350.550	65.55	450.800	75.80
14-01-2020 09:00	350.550	65.55	440.900	76.90
14-01-2020 10:00			500.100	76.10
14-01-2020 11:00			480.110	76.11
14-01-2020 12:00			470.120	76.12
14-01-2020 13:00			460.130	76.13
14-01-2020 14:00			440.140	77.14
14-01-2020 15:00			420.150	77.15
14-01-2020 16:00			430.160	77.16
14-01-2020 17:00	345.170	67.17	445.170	77.17
14-01-2020 18:00	425.180	67.18	525.180	77.18
14-01-2020 19:00	500.190	67.19	600.190	77.19
14-01-2020 20:00	550.200	68.20	650.200	78.20
14-01-2020 21:00	711.210	66.21	700.210	78.21
14-01-2020 22:00	700.220	68.22	800.220	78.22
14-01-2020 23:00	711.230	66.23	850.230	78.23
15-01-2020 00:00	800.240	68.24	900.240	78.24

External: f11 13-01-2020 00:00 GMT



An example of scatter plot configuration that uses all optional elements, together with the resulting plot:

Scatter plot example (click to expand)

```

<displayGroup id="pk" name="Pompkrommes">
  <singleParentLocationDisplays>
    <locationId>261VPrg-
08017</locationId>
    <locationId>261VPrg-
10021</locationId>
    <plotId>pompkromme<
/plotId>

<addInteractionScatterPlot>

<xAxisParameterId>Q.meting.keten<
/xAxisParameterId>

<yAxisParameterId>H.opvoer</yAxisParameterId>
    <dataLine>

<lineColor>pink</lineColor>

<lineStyle>none</lineStyle>
    </dataLine>

<chartTitle>Pompkarakteristieken</chartTitle>

<interactionLine>

<lineColor>red</lineColor>

<lineStyle>solid;thick</lineStyle>

<lineWidth>2</lineWidth>

<labelText>30 Hz</labelText>

<referencePoints>

<xAttributeId>Q_30Hz</xAttributeId>

<yAttributeId>H_30Hz</yAttributeId>
    <
/referencePoints>
    <
/interactionLine>
    <interactionLine>

<lineColor>blue</lineColor>

<lineStyle>solid</lineStyle>

<lineWidth>1</lineWidth>

<labelText>samenloop</labelText>

<referencePoints>
    <point>

<x>600</x>

<y>5</y>
    <
/point>

<point>

<x>700</x>

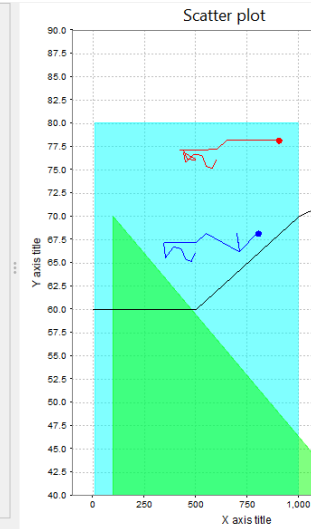
<y>4</y>
    <
/point>

<point>

```

GMT	A		B		C		D	
	Q, rated		H, rated		Q, rated		H, rated	
	(m³/s)		(m)		(m³/s)		(m)	
	Location C		Location C		Location D		Location D	
	LocC		LocC		LocD		LocD	
	[1]		[1]		[1]		[1]	
	13-01-2020		13-01-2020		13-01-2020		13-01-2020	
	00:00		00:00		00:00		00:00	
	Series 1		Series 1		Series 2		Series 2	
14-01-2020 01:00		500.100		66.10		600.100		76.10
14-01-2020 02:00		480.200		65.20		580.200		75.20
14-01-2020 03:00		460.300		65.30		560.300		75.30
14-01-2020 04:00		450.400		65.40		550.400		75.40
14-01-2020 05:00		430.500		66.50		530.500		76.50
14-01-2020 06:00						515.600		76.60
14-01-2020 07:00		390.700		66.70		490.700		76.70
14-01-2020 08:00		350.550		65.55		450.800		75.80
14-01-2020 09:00		350.550		65.55		440.900		76.90
14-01-2020 10:00						500.100		76.10
14-01-2020 11:00						480.110		76.11
14-01-2020 12:00						470.120		76.12
14-01-2020 13:00						460.130		76.13
14-01-2020 14:00						440.140		77.14
14-01-2020 15:00						420.150		77.15
14-01-2020 16:00						430.160		77.16
14-01-2020 17:00		345.170		67.17		445.170		77.17
14-01-2020 18:00		425.180		67.18		525.180		77.18
14-01-2020 19:00		500.190		67.19		600.190		77.19
14-01-2020 20:00		550.200		68.20		650.200		78.20
14-01-2020 21:00		711.210		66.21		700.210		78.21
14-01-2020 22:00		790.220		68.22		800.220		78.22
14-01-2020 23:00		711.230		66.23		650.230		78.23
15-01-2020 00:00		800.240		68.24		900.240		78.24

Externat: [1] 13-01-2020 00:00 GMT



```

<x>1200</x>

<y>8</y>
<
/point>
<
/referencePoints>
<
/interactionLine>
<referencePointsLegendLabel>Q<
/referencePointsLegendLabel>

<coloredBackgroundArea>

<label>werkgebiet0</label>

<color>green</color>

<opaquenessPercentage>25</opaquenessPercentage>
<point>

<x>600</x>

<y>5</y>
</point>
<point>

<x>700</x>

<y>4</y>
</point>
<point>

<x>1200</x>

<y>8</y>
</point>
<

/coloredBackgroundArea>

<coloredBackgroundArea>

<label>werkgebiet1</label>

<color>green1</color>

<opaquenessPercentage>10</opaquenessPercentage>

<xAttributeId>enkelloop_Q</xAttributeId>

<yAttributeId>enkelloop_H</yAttributeId>
<
/coloredBackgroundArea>

<overrulingSeriesLabel>QH<
/overrulingSeriesLabel>

<markerVisibility>all</markerVisibility>
<
/addInteractionScatterPlot>
</singleParentLocationDisplays>
</displayGroup>

```

## statisticalFunctionId

(since 2021.02) Using `statisticalFunctionId` it is possible to create a preconfigured display with any statistical function.



The statistical function should have an id configured in TimeSeriesDisplay.xml, for example :

```
<statisticalFunction function="centralMovingAverage" id="CMA" ignoreMissings="true">
  <movingAccumulationTimeSpan unit="hour" multiplier="6"/>
</statisticalFunction>
```

This id can be used in displays to create a predefined display with the required statistical function :

```
<display name="Central Moving Average">
  <locationId>LocD</locationId>
  <plotId>PlotId</plotId>
  <statisticalFunctionId>CMD</statisticalFunctionId>
</display>
```