09 Report Module

What	*nameofinstance.*xml
Description	Configuration for the Report module
schema location	https://fewsdocs.deltares.nl/schemas/version1.0/reports.xsd

Related pages

- Generate Image from HTML report
- Generate PDF from HTML report
- HyFS report model xml report generation
- Improved report configuration
- Tags in report template
- Using Colours in Tables

- CSS Table Styles

Table of Contents

Report Module Configuration

The role of the report module is to generate reports of the forecast results in a form that can easily be displayed without using DELFT-FEWS on a web server. Reports that are produced by DELFT-FEWS serve two purposes:

- provision of detailed, specific information to e.g. forecasting duty officers, area officers
- provision of general, overview reports

Reports are used to present the forecast result data in a fixed, user defined, format. The format of a report can be easily customized by a user and stored in a template.

Some functions of the DELFT-FEWS use the report component to present the results. For example the critical condition lookup tables defined in for example coastal forecasting only produce index time series. Without post-processing of these indices, a user can never see what is happening in his coastal system. The report component will therefore also be used to interpret the indices and transform these into information that a user can understand. The ValueAttributeMaps (see Regional Configuration) define how these indices are to be transformed to understandable strings and/or icons.

Reports generated can be exported directly to the file system (for viewing through the web server) or they can be stored in the database. The report export module may then be used to exporting selected reports when required to the web server.

The report template uses tags as placeholders to identify the location of objects in the report. In the following table the available tags are described. This page gives an overview of available tags. It also gives details on how defined declarations are to be used in the reports- allowing layout of tables etc. to be defined.

For many report elements, the MC ID should be set in the global.properties in "REPORTS_LEFTCOLMCID" (REPORTS_LEFTCOLMCID=none for Stand Alone).

When available as configuration on the file system, the name of the XML file for configuring an instance of the correlation module called for example Report_Coastal_Forecast may be:

Report_Coastal_Forecast 1.00 default.xml

Report_Coastal_Forecast	File name for the Report_Coastal_Forecast configuration.
1.00	Version number
default	Flag to indicate the version is the default configuration (otherwise omitted).
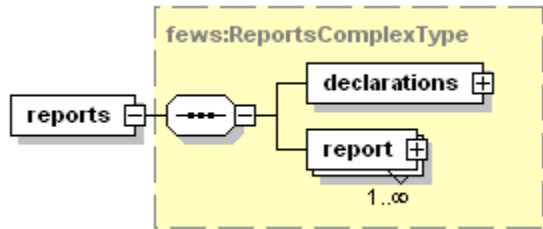
Figure 89 Elements of the Reports configuration

Configuring formatting of reports

declarations

Root element for declaring variables and formats to be used in making the report.

defineGlobal

0..∞

chartFormat

0..∞

summaryFormat

0..∞

tableFormat

0..∞

statusFormat

0..∞

dateFormat

0..∞

numberFormat

0..∞

ReportsDeclarationsComplexTy...

templateDir

reportsRootDir

reportsRootSubDir

sendToLocalFileSystem
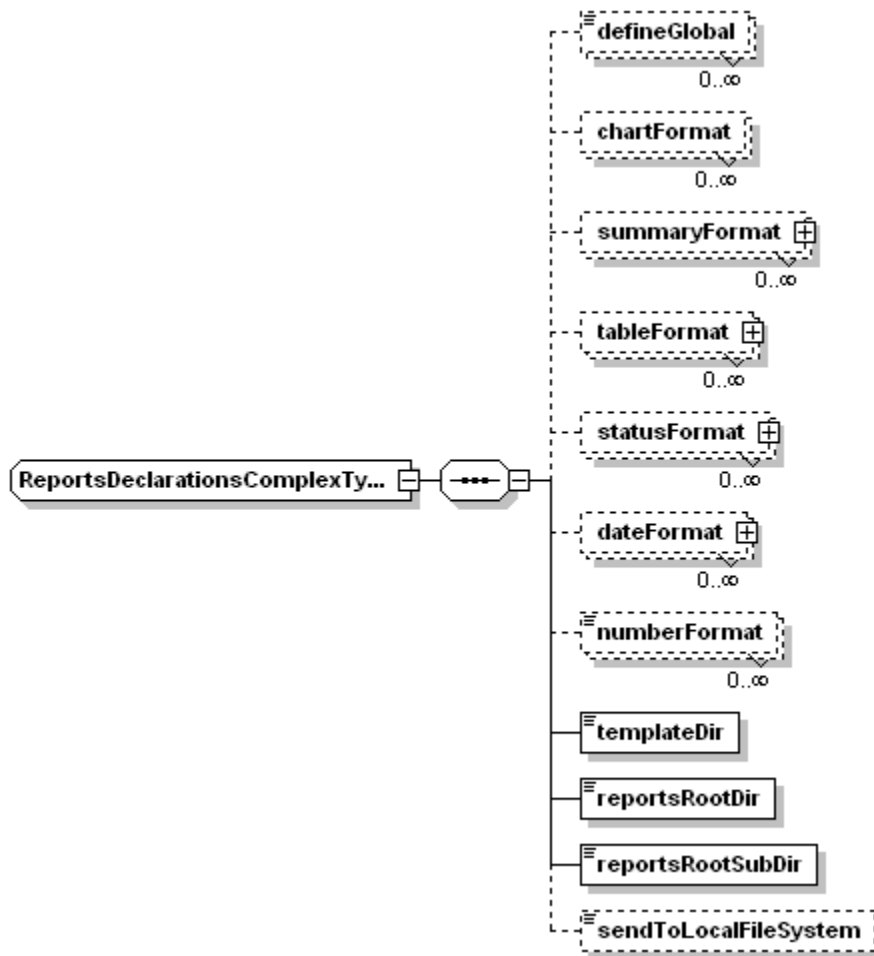
Figure Elements of the declarations configuration

defineGlobal

The DefineGlobal element can be used to enter some information that is valid for all reports. Multiple DefineGlobal elements may be defined, as long as the ID attribute is kept unique.

Related TAG: $DEFINITION(definitionId)$

<item>Format

Formatting instructions for a specific item in the report.

templateDir

Root directory of the directory containing report templates.

reportRootsDir

Root directory to which the all reports are to be exported. This directory is typically the root directory of the web server

Note: A global.properties tag may be useful to define this directory

reportRootsSubDir

Root directory to which the current reports are to be exported. This directory is relative to the reportsRootsDir

sendToLocalFileSystem

Boolean option to determine if reports are to be written to the file system on creation. If set to false reports will be stored in the Reports table in the database, pending export through the ReportExport module.

temporary

When temporary the reports are only available for the report export module executed in the same run. The reports are not saved to the database but instead to a temp file that is deleted at the end of the run

Configuring content of reports

report

Root element for defining the data to be included in the report.

Figure 91 Elements of the Report configuration

InputVariable

Input timeseriesSets for the report. All timeSeriesSets that are used in the report must be defined here, both for carts and for tables. See Transformation Module for details on defining an inputVariable element.

FileResource

Reference to an external file that must be copied with the report. The file will be copied to the same location as the report, with the correct filename inserted at the place of the tag. This can be used for copying images.

Related TAG: $FILERESOURCE(resourceId)$

In the Report configuration file a <fileResource id="Plot1">Plot1.svg</fileResource> need t be added with an id. The fileresource file "Plot1.svg" is a file that need to be included in the \Config\ReportImageFiles\ folder.

In the report template file a TAG need to be included with the fileResource ID: <img src="$FILERESOURCE(Plot1)$">

Chart

In the Chart element the variableId's to be used for one or more charts are defined. The Chart ID that is defined is referenced in the TAG.

Related TAG: $CHART(chartId)$

Summary

In the Summary element the variableId's are specified that are used to create the summary information. The OverlayFormat of the SummaryFormat determines what is shown on the map.

Related TAG: $SUMMARY(summaryId)$

Table

In the Table element the variableId's are specified that are used to create a table. The TableFormat controls how the table is formatted, i.e. the row and column information and how the data is displayed in the map.

Related TAG: $TABLE(tableId)$

See for more information Tables (Overview) below.

Status

The Status element links a Status ID that is referenced in the STATUS TAG to a Status Format ID.

Related TAG: $STATUS(statusId)$

Template

Template file name to be used for the report.

OutputSubDir

Optional output subdirectory for this report.

If singleLocation is true, then outputSubDir can contain location attribute ids, e.g. "H_ (BASIN)", where tags between "@" signs refer to location attributes that are defined in the locationSets config file. These tags will be replaced with the corresponding attribute values of the location of the report. These values can be different for different locations. See 22 Locations and attributes defined in CSV files, Shape-DBF files or external tables for more information. If required to use a "@" sign in this element without using locationAttributes, then need to add an extra escape character "@" before each "@" sign. In other words "@@" is replaced with a single "@" sign.

OutputFileName

Output file name for this report.

The following tags can be used to customize the file output name:

- %LOCATIONID%
- %TIMESERIES_POSTFIX(dateFormatId)%
- %CURRENTTIME(dateFormatId)%
- %TIMEZERO(reportVariableId;dateFormatId)%
- %TIMESTEP(dateFormatId)% (only supported for spatialPlotSnapshots snapshot fileNames).

%LOCATIONID %

If singleLocation is true, then can use tag %LOCATIONID% here, which will be replaced with the id of the location of the report. If singleLocation is true, and the %LOCATIONID% tag is not present, then a "_" sign and the id of the location of the report will be inserted automatically just before the "." sign in the filename.

Additionally, if singleLocation is true, then outputFileName can contain location attribute ids, e.g. "H_ (BASIN)", where tags between "@" signs refer to location attributes that are defined in the locationSets config file. These tags will be replaced with the corresponding attribute values of the location of the report. These values can be different for different locations. See 22 Locations and attributes defined in CSV files, Shape-DBF files or external tables for more information. If required to use a "@" sign in this element without using locationAttributes, then need to add an extra escape character "@" before each "@" sign. In other words "@@" is replaced with a single "@" sign.

%TIMESERIES_POSTFIX(dateFormatId)%

The TIMESERIES_POSTFIX tag will be replaced by the minimum date of the configured timeSeries in the report. If no timeSeries have been configured, the system date will be used. An optional dateFormat id can be specified. For example: %TIMESERIES_POSTFIX(dateFormatId)% where dateFormatId is the id of a dateFormat as specified in the reports declarations. If no dateFormatId is passed, the default date format ddmmyyyym will be used.

%CURRENTTIME(dateFormatId)%

The CURRENTTIME tag will be replaced by the current system time. An optional dateFormat id can be specified (see TIMESERIES_POSTFIX).

%TIMEZERO(reportVariableId;dateFormatId)%

The TIMEZERO tag will be replaced by the minimum T0 of the timeSeries configured by the reportVariableId. If no parameter is configured, T0 of the task run will be used. The reportVariableId is optional.

An optional dateFormat id can be specified (see TIMESERIES_POSTFIX).

N.B.: Be careful in specifying date formats that are compatible with the filesystem. For example and : in the dateformat will not be compatible with the windows filesystem.

%TIMESTEP(dateFormatId)%

The TIMESTEP tag will be replaced by the timestep of a spatialPlotSnapshots snapshot. An optional dateFormat id can be specified. For example

```
<snapshot id="Waterlevel_0">
        <relativeTime unit="hour" value="0"/>
        <fileName>Report.EAM.Waterlevel.FC_%TIMESTEP(df1)%</fileName>
</snapshot>
```

DefineLocal

The DefineLocal element can be used to enter some information that is valid for a single report. Multiple DefineLocal elements may be defined, as long as the ID attribute is kept unique.

A special case of the DefineLocal attribute is when the DefineLocalID is the same as a previously DefineGlobalID. In this case the DefineLocal overrides the setting of the DefineGlobal. This is valid only for the report being configured, not for any other configured reports in the same configuration file.

Related TAG: $DEFINITION(definitionId)$

The template function tag $DEFINITION(defineLocalId)$ will be replaced with the value of the defineLocal element with the corresponding id.

If singleLocation is true, then the value of a defineLocal element can contain location attribute ids, e.g. "H_ (BASIN)", where tags between "@" signs refer to location attributes that are defined in the locationSets config file. These tags will be replaced with the corresponding attribute values of the location of the report. These values can be different for different locations. See 22 Locations and attributes defined in CSV files, Shape-DBF files or external tables for more information. If required to use a "@" sign in this element without using locationAttributes, then need to add an extra escape character "@" before each "@" sign. In other words "@@" is replaced with a single "@" sign.

Charts

Charts can be used for visualizing timeseries. The charts which can be added to html-reports look more or less the same as in the TimeSeries Display. The configuration is simular to display groups. Charts are being created as (individual) *.png files.

Figure 92 Example of a chart (*.png file)

Template tag

In the Chart element the variableId's to be used for one or more charts are defined. The Chart ID that is defined is referenced in the TAG.

Related TAG: $CHART(chartId)$

Configuration aspects

Chart should be configured according the following schema definition.

ChartReportChartComplexType

fews:ChartComplexType (extension)

attributes

- fews:title
- fews:backgroundColor
- fews:backgroundOpaquenessP...
  - fews:plotBackgroundColor
  - fews:plotBackgroundOpaquene...
- fews:plotBackgroundReportIma...
- fews:plotForegroundReportIma...
  - fews:leftAxisTickUnit
  - fews:leftAxisTick 1..∞

    Deprecated. Use leftAxis to overrule tickUnit or ticks
  - fews:rightAxisTickUnit
  - fews:rightAxisTick 1..∞

    Deprecated. Use rightAxis to overrule tickUnit or ticks
- fews:leftAxisScaleUnit

  Deprecated. Use leftAxis to overrule scaleUnit
- fews:rightAxisScaleUnit

  Deprecated. Use rightAxis to overrule scaleUnit
- fews:leftAxis

  Use leftAxis to overrule settings for the left axis in the chart format
- fews:rightAxis

  Use rightAxis to overrule settings for the right axis in the chart format
- fews:forecastConfidenceTime...
- fews:forecastConfidenceTime...
- fews:forecastConfidenceTime...
- fews:thresholdGroupId

fews:thresholdGroupId

Optional. If specified, then only thresholds in the specified thresholdGroup are displayed in this chart. If not specified, then thresholds in all thresholdGroups can be displayed.

fews:area

0..∞

Displays the extent of multiple time series as a single area. The area between the max

and min values for each time are filled with a single color. The areas are always displayed behind the other lines.

The area is painted on top of areas already specified, so specify the largest areas first.

fews:timeSeries

0..∞

fews:fileName

File name for this chart. If singleLocation is true, then can use tag %LOCATIONID% here, which will be replaced with the id of the location of the report. If singleLocation is true, and the %LOCATIONID% tag is not present, then a "_" sign and the id of the location of the report will be inserted automatically just before the "." sign in the filename.

fews:fileFormat

Image format of the chart. PNG is the default.

Generated by XMLSpy   www.altova.com

Figure 93 Chart definition according the ChartComplexType (reports.xsd).

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 assigned 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 assigned and an optional right y axis to which a timeseries or rating curve is assigned.

Both left y axis and right y axis can be customized using <leftAxis> and <rightAxis> configuration elements. Note that <leftAxis> configuration is used only if at least one time series is attached to the left axis. The same goes for the <rightAxis>.

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. Note that further constraints on what thresholds are displayed is possible in the reports schema.

Plot background and foreground images

The images configured in <plotBackgroundReportImage> and <plotForegroundReportImage> should be included in the config folder ReportImageFiles. The images are resized to fill the whole plot. The foreground image should be transparent. Below an example of the plot with the background image.
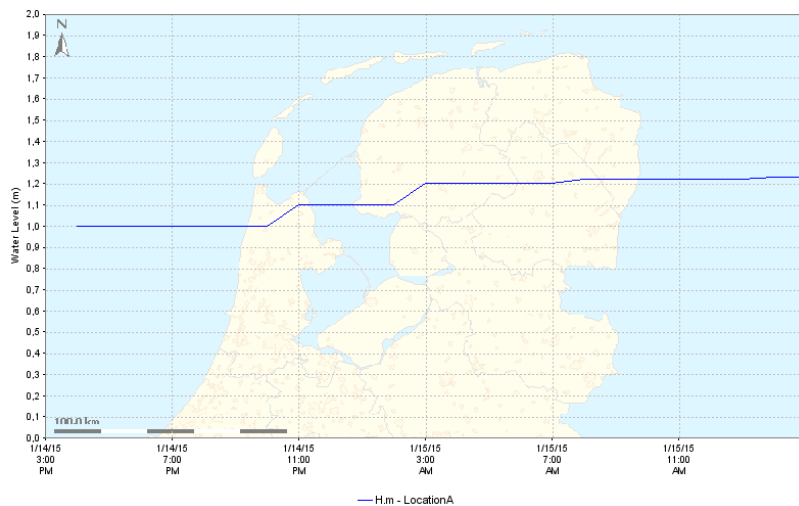
Multiple TimeSeries plots

To assign a timeseries to the left or right y axis use the <axis> element. This element is optional and leaving it out will assign 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.

```
<chart id="
Example7" formatId="
chartFormat2" width="500"
height="306">

<timeSeries preferredColor="
green" lineStyle="solid;
thick" axis="right"
visibleInLegend="true"
label="Gemessen">series1<
/timeSeries>

<timeSeries preferredColor="
red" lineStyle="solid;
thick" axis="right"
visibleInLegend="true"
label="COSMO2">series2<
/timeSeries>

<timeSeries preferredColor="
orange" lineStyle="solid;
thick" axis="left"
visibleInLegend="true"
label="COSMO7">series3<
/timeSeries>

<timeSeries preferredColor="
violet" lineStyle="solid;
thick" axis="left"
visibleInLegend="true"
label="ECMWF">series4<
/timeSeries>

<fileName>Example7<
/fileName>
</chart>
```
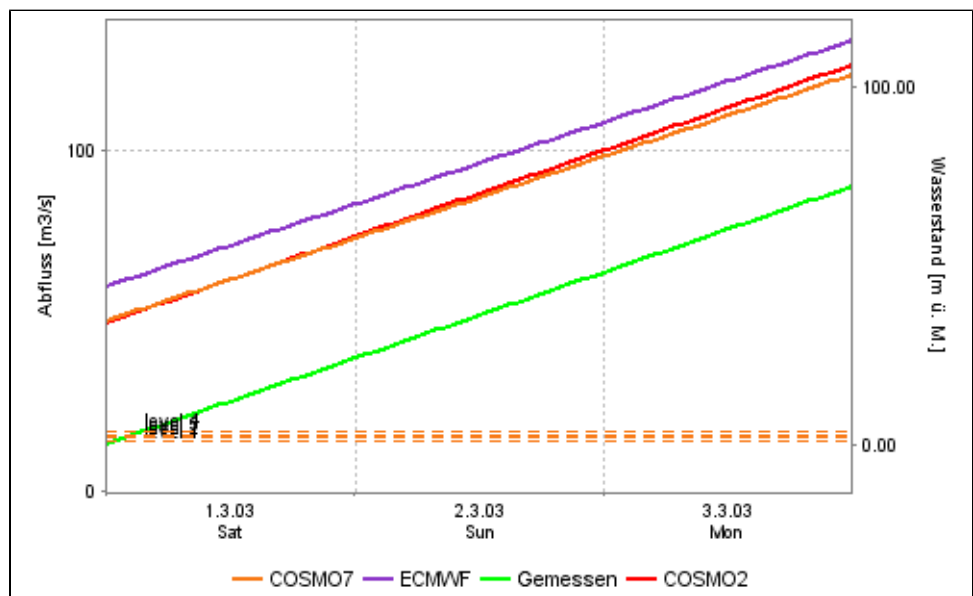

Threshold line only for timeseries on the left y axis

Stage/discharge plots

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.

Stage/discharge plots support both PI rating curves and deprecated rating curves (deprecated rating curves are ratings obtained from xml file according to the schema ratingCurves.xsd)

Below an configuration and resulting plot. The option reverse is applicable only to deprecated rating curves.

```
<chart id="Example4"
formatId="chartFormat1" width="900"
height="300">
                <rightAxis>

<ratingCurve>

<ratingCurveId>location1</ratingCurveId>

<reverse>true</reverse>
                        <
/ratingCurve>
                </rightAxis>
                <timeSeries
axis="left" preferredColor="blue"
markerStyle="none" lineStyle="solid;
thick" label="geen verwachting"
visibleInLegend="true">series1<
/timeSeries>

<fileName>Example4</fileName>
        </chart>
```


Threshold line only for timeseries on the left y axis
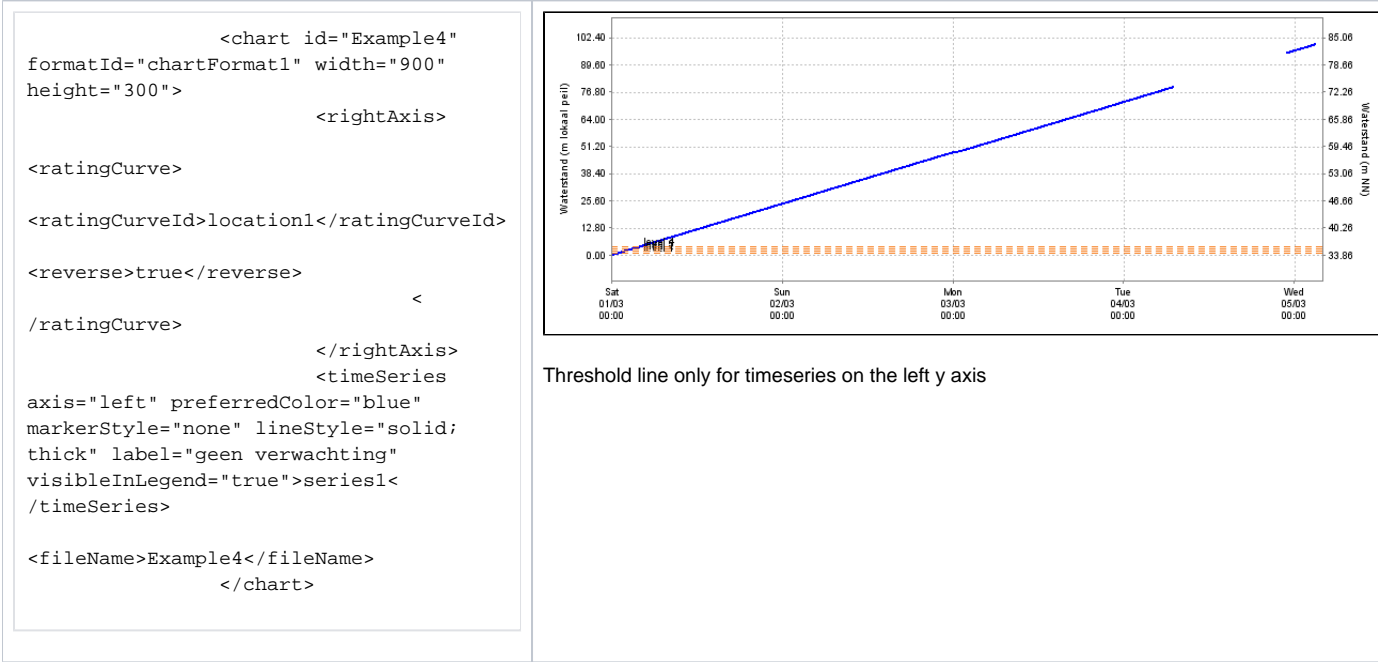
```
            <chart id="Example8"
formatId="chartFormat1" width="900"
height="300">
                    <rightAxis>

<ratingCurve>

<ratingCurveId>location1</ratingCurveId>

<reverse>true</reverse>
                                    <
/ratingCurve>
                    </rightAxis>
                    <timeSeries
axis="both" preferredColor="blue"
markerStyle="none" lineStyle="solid;
thick" label="geen verwachting"
visibleInLegend="true">series1<
/timeSeries>
                    <timeSeries
axis="left" preferredColor="black"
markerStyle="circle" lineStyle="solid;
thick" label="meetwaarde" markerSize="3"
>series2</timeSeries>

<fileName>Example8</fileName>
            </chart>
```
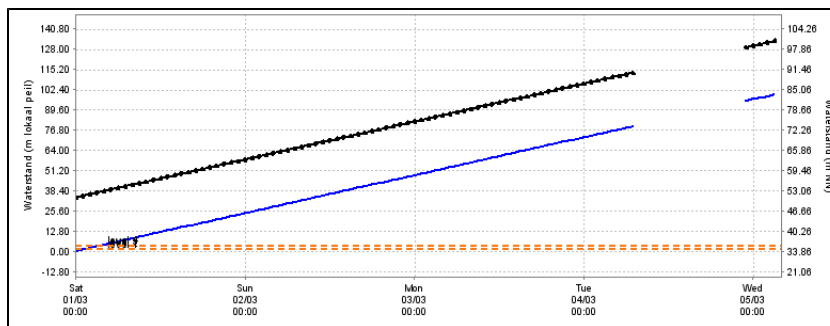
Threshold line only for timeseries on the left y axis

For adding a chart to a report the following aspects are important to configure:

- Chart attributes;
- Chart timeseries;

Chart attributes

To display the chart the following attributes need to be defined:

- Id: unique identifier for this chart (unique in this report);
- formatId: referring to the format of this chart (in declaration section of the report)
- height: an integer value for the height of the chart in pixels;
- width: an integer value for the width of the chart in pixels;

Chart timeseries

To display lines, dots etc. of a certain timeseries, the reference to this timeseries (variableId) should be mentioned.

Figure 94 chart declaration in the report section

Rotated Markers

Since 2017.02 FEWS supports the possibility to plot arrows markers showing direction on plots, which for example can be used to create a plot of wind speed where the arrow markers signify the wind direction. The report module also supports such markers. Configuration for these markers must be included in the TimeSeriesDisplayConfig.xml, as described in the Time Series Display Configuration Guide under the "Directional arrow markers" section. In addition, for the direction time series to be available in the report module, the direction time series must be included in the chart in addition to the speed time series. The direction time series is NOT added automatically.

Direct reference to existing display ID

A new feature in FEWS version 2021.01 is the option to create a chart using a direct reference to an existing Display ID in the DiplayGroups configuration. You can configure this using the <DisplayChart/> element like this:

```
<displayChart id="testDisplayChart" width="400" height="400">
    <displayId>testDisplay</displayId>
 </displayChart>
```

Using a classic <chart/> tag you have to reconfigure each chart separately for both the TimeSeriesDisplay on screen and the Report Export. The Benefit of this may be that you can optimize the way the charts look in a printed or web based report, which may have a different resolution, fonts and other features relative to the computer screen. However, if you just want the charts in the report to look the same as they do on the screen, using the existing Display ID in a <displayChart /> element can save you a lot of work. Optionally you can specifiy a fileName and fileFormat, the default output will be named chart00. png, chart001.png etc.

Spatial plot snapshots

Gridded time series can be visualized in a report by means of snapshots. The snap shot is an image depicting the time series spatially.

Configuration aspects

Spatial plot snapshots are configured according to the following schema definition (sharedTypes.xsd). Since FEWS 2023.01 multiple class breaks are allowed. When multiple class breaks are configured in the best suitable class breaks are automatically chosen

```
<displayChart id="testDisplayChart" width="400" height="400">
```

fews:AnimationComplexType (extension)

attributes

id

name
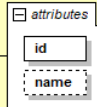
fews:**description**

fews:**mapExtentId**

Since 2017.02.
FEWS-17986. When this
gridPlot is selected, the zoom
extent will be changed to the
given map extent. Can also
be specified through defaults.

fews:**displayGroupPlotId**

When double clicking on a location
in the spatial display the time
dialog will be displayed with the
display group for the specified plot
and the double clicked

location.

fews:**dataFeedId**

Since 2019.02. Supported
only for external historical
time series, in online
systems only (no
stand-alone). If configured,
the spacial grid display will
display in the footer when
the latest import happened
for the given dataFeedId.

fews:**timeSeriesSet**

1..∞

fews:**valueTimeSeriesSet**

fews:**ObsoleteTimeSeriesSetChoice**

fews:**directionTimeSeriesSet**

1..∞

fews:**uTimeSeriesSet**

fews:**vTimeSeriesSet**

1..∞

fews:**dataLayer**

Use this option to display
multiple data grids. Each grid
will be displayed in a
separate data last, will be
shown on top. If a layer

contains

a
timeSeriesSet with scalar
data (possibly with a
locationSet with multiple
locations), then this will be
converted to a grid to
display in that layer.

fews:**quadtreeWaterLevelsLayer**

Since 2013.01. FEWS-8729. Use this option to
display water level data that is defined on a
quadtree grid. The water levels will be shown
using the colours configured

in the classBreaks. This only works if
a local datum reference layer (e.g.
coverageTileArchiveLayer) has been defined in
the corresponding geoMap with digital elevation
model

(DEM) data.

fews:**QuadTreeLayerSequence**

fews:**quadtreeSubgridVelocitie...**

Since 2013.01. FEWS-8729. Use this option to
display flow velocity data that is defined on the
flow links of a quadtree grid. The specified
timeSeriesSet should

contain speed values normal to the
quadtree cell faces for each flowlink. A flowlink
is the "connection" between two adjacent
quadtree cells. This only works if the specified

flow velocity data comes from the
same quadtree grid as the corresponding
quadtreeWaterLevelsLayer and if a local datum
reference layer (e.g.
coverageTileArchiveLayer) has been

defined in the corresponding geoMap
with digital elevation model (DEM) data. The
velocities within the quadtree cells are
computed from the velocities on the flowlinks
using

the subgrid method (see the paper
"Quadtree flood simulations with sub-grid digital
elevation models" by Guus S. Stelling that is
available from

fews:**AnimatedLayerChoice**

0..∞

Since 2018.01. Omit the layers
only if the grid plot is used as a
template (format) . Then the time
series should be injected by the
responsible components.

http://www.3di.nu/wp-content/uploads/2013/02/Stelling_ice_2012.pdf). The velocities are shown using arrows that use the symbolSizes configured in the arrowClassBreaks.

fews:trackLayer ⊞

Use this option to display a cyclone track. The whole track is always visible. The point for the time in the slider is highlighted.

fews:wmsLayer ⊞

Since 2013.01. FEWS-8875. Use this option to display an (animated) wms layer.

fews:logo ⊞

Since 2016.02 #68240. (Copy right logo) visible in the map. Please note that the logo cannot be shown in the maps of the Report module

fews:showThumbnailsPanel

If this is true, then the grid display shows the spatial thumbnails panel when this grid plot is selected. Otherwise the grid display will not show the spatial thumbnails panel when this grid plot is selected. Default is false. After this grid plot has been selected, the user can still switch on/off the spatial thumbnails panel manually, using the toggle button in the grid display. If this element is never configured for any grid plot, the user settings will be used to decide whether the thumbnails panels is shown on start-up. If this element is configured for any grid plot, the thumbnails panel will always be hidden on start-up.

fews:numberOfRecentForecasts

The grid thumbnails panel in the grid display shows the specified number of most recent forecasts with a time0 before or at the display time. It shows one row of thumbnails for each forecast shown. This option is only used for gridPlots that show forecast time series. For gridPlots that do not show forecast time series, only one row of thumbnails is shown. Default is 1, i.e. by default thumbnails are only shown for the current forecast.

SpatialPlotSnapshotsComplexT...

fews:gapBetweenThumbnails

The gap between two consecutive thumbnails in the grid thumbnails panel (in number of timeSteps). If this is 1, then a thumbnail is shown for every timeStep. If this is 2, then a thumbnail is shown for every other timeStep. If this is 3, then a thumbnail is shown for each third timeStep, etc. Default is 1. The first time in the displayed period is the reference time where the gaps "start counting", i.e. there is always a thumbnail for the first time in the displayed period.

fews:movieFrameDurationMillis

The number of milliseconds a frame / time step is visible before the next time step becomes visible.
                    The default is 200 milliseconds.
                    When the CPU is too slow to catch the specified frame rate a frame will be displayed longer than specified.

fews:movingAverageTimeSpan ⊞
1..∞

When configured a slider is displayed with the configured time spans. If a time span is selected in the slider, then the data in the grid display is aggregated (moving average) using that time span. The start of a timeSpan is exclusive and the end of a timeSpan is inclusive. Configure a very large time span (eg 100000 days) to allow the user to see the total average of the loaded time series.

fews:movingAccumulationTim... ⊞
1..∞

When configured an accumulation slider is displayed with the configured time spans. If a time span is selected in the slider, then the data in the grid display is aggregated (moving accumulation) using that time span. The start of a timeSpan is exclusive and the end of a timeSpan is inclusive. Configure a very large time span (eg 100000 days) to allow the user to see the total sum of the loaded time series.

fews:accumulationTimeSpan ⊞
1..∞

Since 2014.02. When configured an accumulation slider is displayed with the configured time spans. When a time span is selected with the slider, the interval of the ticks in the time slider in the grid display is recalculated to the selected time step. Configure a very large time span

(eg 100000 days) to allow the user to see
the total sum of the loaded time series.

fews:accumulationTimeStep ⊞

1..∞

Since 2016.02. When configured an
accumulation slider is displayed with the
configured time steps. When a time step
is selected with the slider, the interval of
the ticks in the time slider in the grid
display is recalculated to the selected time
step. Configure a very large time step
(eg 100 years) to allow the user to
see the total sum of the loaded time series.

fews:hideLastValueCheckBox

Since 2017.02. Default false. When set to
true, the "show last value" will be hidden.
Note that if an accumulation slider is
configured, the checkbox will be hidden by
default.

fews:locationFilterAttributeId

0..∞

Specify the location attributes the user
can use to filter the
locations/polygons/lines that are visible in
the plot. When specified the filter by
location attribute button will be enabled.
If a attribute name besides the id in the
locationsSets.xml is specified this will be
used in the attribute filter tree. The
feature will not work for grid layers

fews:GridPlotLegendChoice

Legend style

fews:legendStyle

Table is default legend. Bar
legend shows fluent scale
bar on the right. If you wish
more advanced bar legend,
use element barLegend

fews:barLegend ⊞

fews:modifiedColor

Since 2019.02. Color used to
mark time periods where a
spatial modifier is applied.

fews:ClassBreaksChoice

0..∞

Optional when default class
breaks are available for this plot
group and parameter group.
Note: the missingValueColor
configured in this classBreaks
element is also used to display
missing values (NaN values) for
point locations and tracks when
the display is showing threshold
warning level flags. Since
2017.01 it is possible to
configure multiple class breaks.
When multiple class breaks are
configured the user can switch
between different class breaks.
This first one is the default. The
configured class break
description is listed in a drop
down box above the class
breaks

fews:classBreaksId

Reference to class breaks
configured in
TimeSeriesDisplayConfig.xm
l.

fews:classBreaks ⊞

Class breaks inlined

fews:ArrowClassBreaksChoice

Use different class breaks for the
arrows. Makes sense when you have
an arrow layer on top.

fews:arrowClassBreaksId

Reference to class breaks configured
in TimeSeriesDisplayConfig.xml.

fews:arrowClassBreaks ⊞

Class breaks inlined

fews:absoluteDifferenceClassB...

Since 2020.02. Class breaks that ares used to
show the difference between two grids

fews:relativeDifferenceClassBr...

Since 2020.02. Class breaks that ares used to
show the relative difference between two grids

fews:contourLinesColor

When specified a contour line with
the class break value is painted for
every new class break.
ONLY
SUPPORTED FOR REGULAR
GRIDS

fews:verticalSliderRange ⊞
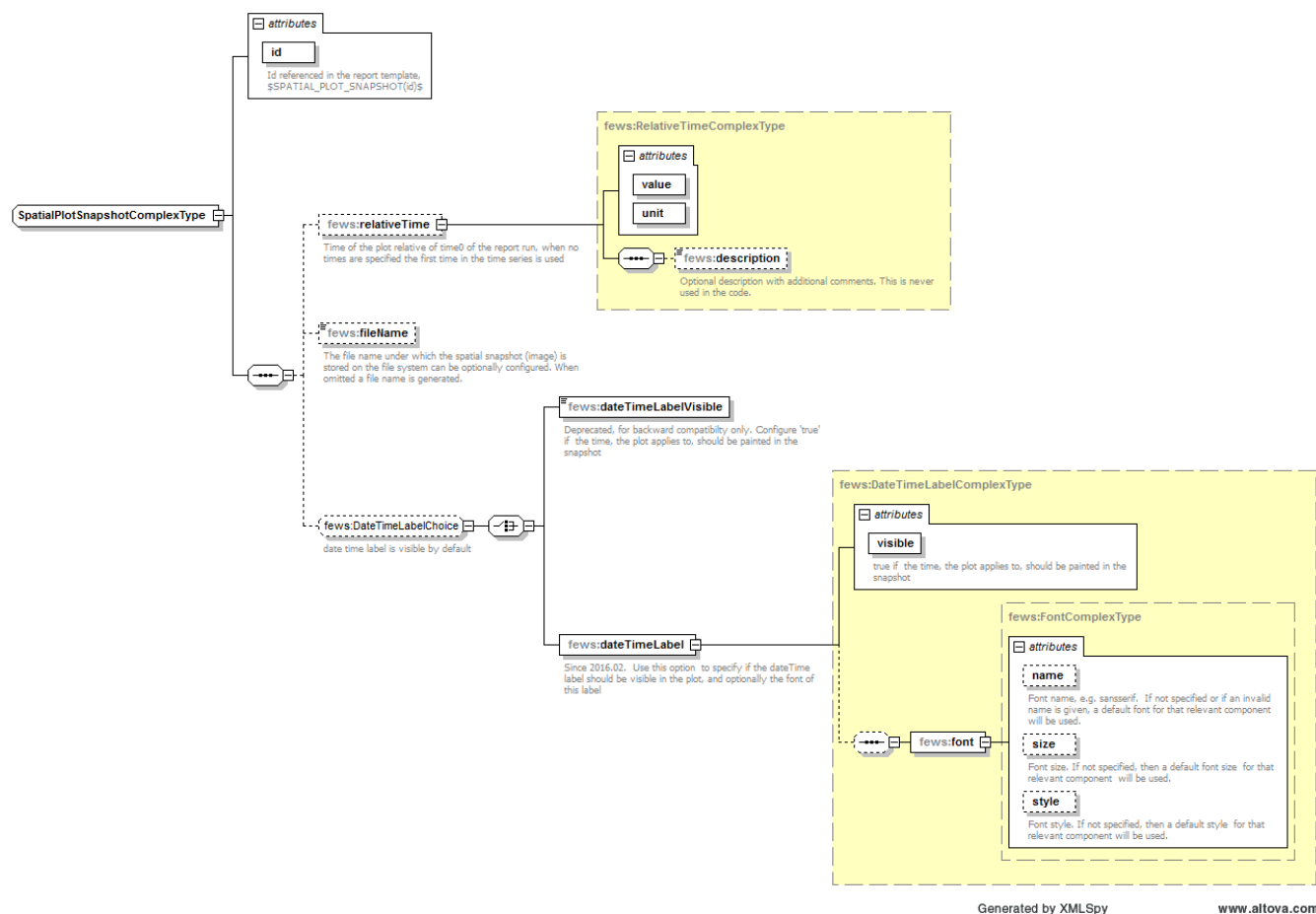
Since 2017.02. Min and max depth
or levels

fews:geoMap ⊞

Optional when a default
map is available for this plot
group and parameter
group

fews:workflow ⊞

fews:computeMapExtentToSho... ⊞

Since 2017.02. If this is configured, then the
default map extent will be enlarged to show all
locations with non-missing time series. The
default map extent is the minimum extent

default map extent is the minimum extent

fews:width

Width of the animation in pixels

fews:height

Height of the animation in pixels

fews:ValueLabelsComplexType

fews:valueLabels

Since 2016.02, optional. If this is configured, then labels will be drawn next to the locations on the map in the produced snapshot/animation. These labels will show the actual values of the corresponding locations and optionally also the units. Please note that labels will not be drawn if they would overlap other labels (decluttering), therefore some locations may not have a label on the map.

fews:showUnits

If this is true, then the units are shown next to the values in the labels. If this is false, then the units are not shown.

fews:textLabels

Since 2018.01, optional. If configured then the location attribute will be drawn next to the locations on the map in the created snapshot/animation

fews:labelFont

Since 2022.01, optional. The font of the valueLabels and textLabels that will be drawn next to the locations on the map in the created snapshot/animation

fews:SpatialPlotSnapshotComplexType

attributes

id

Id referenced in the report template. $SPATIAL_PLOT_SNAPSHOT(id)$

fews:snapshot

1..∞

Snapshot

fews:relativeTime

Time of the plot relative of time0 of the report run, when no times are specified the first time in the time series is used

fews:fileName

The file name under which the spatial snapshot (image) is stored on the file system can be optionally configured. When omitted a file name is generated.

fews:DateTimeLabelChoice

date time label is visible by default

fews:dateTimeLabelVisible

Deprecated, for backward compatibility only. Configure 'true' if the time, the plot applies to, should be painted in the snapshot

fews:dateTimeLabel

Since 2016.02. Use this option to specify if the dateTime label should be visible in the plot, and optionally the font of this label

Generated by XMLSpy   www.altova.com

snapshot

The snapshot is defined as a relative time interval from T0. Optionally a file name may be specified for the snapshot which is used to save the snapshot on the file system. If omitted the file name is generated by the report module. If required the date-time label can be adjusted (font type and font size)

Generated by XMLSpy   www.altova.com

```
<spatialPlotSnapshots id="spatialPlotSnapshot">
.......
......
  <snapshot id="Snapshot">
      <relativeTime value="0" unit="hour"/>
      <fileName>SnapshotFile</fileName>
      <dateTimeLabel visible="true">
            <font name="SansSerif" size="12" style="bold"/>
      </dateTimeLabel>
  </snapshot>

</spatialPlotSnapshots>
```

Summary

A summary is a (background) map which can be added to a report. On top of this map, icons can be displayed. The icons indicate the location or the (warning) status of the monitoring or forecasting point. By adding specific html-functionality to the report template(s), maps can be used to navigate through the reports as well. Clickable areas and locations can be distinguished here. The map itself (as a file) can be any image file. For displaying maps in html-reports the following formats are advised: *.png, *.jpg or *.gif.

Template tag

In the Summary element the variableId's are specified that are used to create the summary information. The OverlayFormat of the SummaryFormat determines what is shown on the map.

Related TAG: $SUMMARY(summaryId)$

Creating a summary

The map itself is an existing file and can be created in several ways. An image processing software package (like Paint Shop Pro) can create a 'screendump' from the map section of The FEWS-Explorer. The FEWS-Explorer itself has some hidden features which can be used as well. The [F12] button can be used for:

- Copy current map to a .png file ([F12]+ L);
- Copy current map extent to the clipboard ([F12]+ K);

The *.png file is named "currentmap.png" and can be found in the /bin directory of your system. The map extent (rectangle containing real world coordinates) can be pasted into any application by choosing Edit-Paste or [Ctrl]+ V. These four coordinates describing the extent of your map picture in world coordinates are needed in the declarations section of the report ModuleConfigFile where you declare this summary.

Remark: Every time you use the above mentioned [F12] features, the png file in the /bin directory and the clipboard is overwritten! In making series of maps you should copy/rename your png file after using this option. You should also paste the map extent in a text editor or spreadsheet directly before repeating the operations with another map extent in the FEWS-Explorer.

Configuring a summary

Declaration section

In the declarations section of the report ModuleConfigFile, the summaryFormat needs to be declared. The following elements should be specified (see figure).

- Id: unique identifier (as reference to this summary)
- Map
  o Image
- file: relative reference/path to the image file;
- width: width of the image file in pixels;
- height: height of the image file in pixels;
  o x0: horizontal margin/shift of the map on the html page;
  o y0: vertical margin shift of the map on the html page;
  o leftLongitude: left side of the map in real world coordinates;
  o rightLongitude: right side of the map in real world coordinates;
  o bottomLatitude: bottom side of the map in real world coordinates;
  o topLatitude: top side of the map in real world coordinates;
  o mapFormat: details for positioning and behaviour of the map;
  o overlayFormat: details for positioning and behaviour of icons on the map;
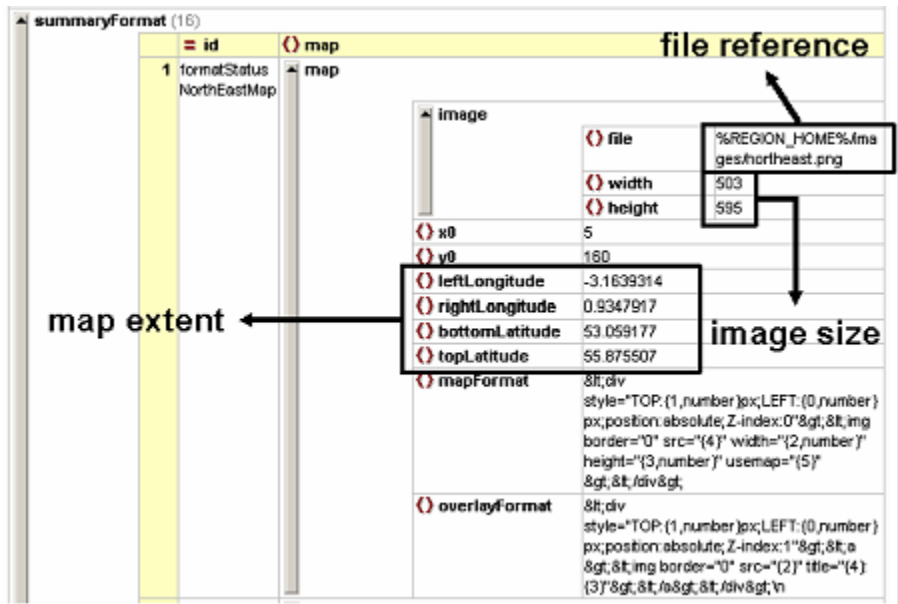
Figure 95 Summaryformat in the declarations section

Detailed explanation

- File details like width and height can be retrieved using image processing software.
- The x0 and y0 elements are margins/shifts of the position of the map compared to the left-upper point of the report (e.g. an A4-sheet). This left-upper point is (0,0). The x0/y0 refer to the left-upper point of the image.
- The mapFormat is used for positioning the map on the page (relative to all other content) and therefor it is placed in a so-called [DIV] tag. This type of html tag puts everything which is configured within that tag in that position on the page. The following table explains the references of the number in this format:

Position	Type	Variable

0	number	Absolute x position of map image.
1	number	Absolute y position of map image.
2	number	Image width.
3	number	Image height
4	number	Image filename.
5 (optional)	number	Reference to a clickable map (#clickmap by default)

- The overlayFormat is used for positioning location and/or warning icons on the map based on:
- The map itself (defined in mapFormat);
- X and Y coordinates of the required locations from the Locations.xml;
- Note that in case of numbers bigger than 999, it can be that the output may contain unwanted commas, e.g. width 1274 is printed in the reports as *1,274*. In order to avoid this, it is recommended to use

```
{<position>, number, #}
```

instead of

```
{<position>, number}
```

.
- Note that when providing a single quote within a snippet of javascript of OverlayFormat in a format field of the report XML configuration and the report appears to be missing this single quote character, this can be fixed by adding the additional escape character in the form of an additional single quote. A more verbose alternative is to put '' or ''.

Position	Type	Variable
0	number	Absolute x position of status icon.
1	number	Absolute y position of status icon.
2	string	Image filename.
3	string	Status description.
4	string	Location name.
5 (optional)	string	Html-link for opening a individual html page (forecastlocation.html)

The mapFormat and overlayFormat elements are configured in html language in which some parts needs to be dynamically filled in. The result will be valid html. The map itself is placed as the bottom 'layer' using the <DIV> tag attributes (index="0") in the mapFormat. Objects defined in the overlayFormat are given higher indices to place them 'on top of the map' and will always be visible (location icons, warning icons).

Use of optional settings in mapFormat and overlayFormat

In both formats, an optional item can be specified.

Clickable map

A clickable map is an image on a html page containing 'hot spots' where the cursor change from the normal 'arrow' to a 'pointing finger' during a *mouse-over* event. These hot spots can contain hyperlinks to other pages.

mapFormat	<div style="TOP:{1, number, #}px; LEFT:{0, number, #}px; position:absolute;Z-index:0"> <img border="0" src="{4}" width="{2,number,#}" height="{3, number, #}" usemap="{5}"> </div>

When adding the string "usemap="{5}" to the mapFormat (see above) the outcome in the html page will be (printed in bold).

The part describing the hot spots for this map are defined in the [map] tag. In this example below, three areas are 'clickable'. Every hot spots links to another html page in the set of reports.

```
<img border="0" src="northeast.png" width="503" height="595" *usemap="#clickmap"*>

<!-- Here the clickable map starts: SHOULD BE IN THE TEMPLATE -->
<map name="clickmap">
<area alt="Northumbria Area" href="northumbria/area_status.html" shape="polygon" coords="138, ...,...,34">
<area alt="Dales Area" href="dales/area_status.html" shape="polygon" coords="266, ..., ..., 285">
<area alt="Ridings Area" href="ridings/area_status.html" shape="polygon" coords="359, ..., ..., 379">
</map>
```

To avoid hot spots on a map, do not include the "usemap="{5}" in the mapFormat.

Hyperlinks

Hyperlinks can be added to the overlayFormat. By using the following option, hyperlinks to individual reports will be added automatically. They will have a fixed (non-configurable) name, "forecastlocation.html" and assuming it is located in a directory with the same name as the locationId compared to where this report with the map is located in the directory structure.

overlayFormat	<div style="TOP:{1,number,#}px; LEFT:{0,number,#}px; position:absolute;Z-index:1"> <img border="0" src="{2}" title="{4}: {3}"> </div>\n

When adding href="{5}" to the overlayFormat (at that location) a hyperlink is being added to the icon placed on the map. In html it will look like this.

```
<!-- map -->
<div style="TOP:160px;LEFT:5px;position:absolute;Z-index:0">
<img border="0" src="derwent.png" width="503" height="578" usemap="#clickmap">
</div>

<!-- icons -->
<div style="TOP:467px;LEFT:427px;position:absolute;Z-index:1">
<a href="BOYNTN1/forecastlocation.html"><img border="0" src="fluvial_site_data.gif"
title="Boyntn1: No threshold exceeded"></a>
</div>
```

The z-index of a icon is determined by the severity of the thresholdWarningLevel that has been crossed, meaning that the locations with the highest crossed thresholdWarningLevel will always be visible (and not potentially hidden behind other icons). Also, future/predicted crossings have a higher z-index then past/observed crossings. The exact Z index for a historical crossing is equal to the severity level. For a forecast crossing it's equal to the maximum configured severity level plus the severity level.

The overlayFormat element can also contain location attribute ids, e.g. " (BASIN)", where tags between "@" signs refer to location attributes that are defined in the locationSets config file. These tags will be replaced with the corresponding attribute values for each location. These values can be different for different locations. See 22 Locations and attributes defined in CSV files, Shape-DBF files or external tables for more information. If required to use a "@" sign in this element without using locationAttributes, then need to add an extra escape character "@" before each "@" sign. In other words "@@" is replaced with a single "@" sign.

Report section

A summary can be added to a report configuration as shown below.

Figure 96 Summary configuration in the reports section

The following elements need to be defined.

- Id: identifier referring to the tag in the template. In this case the corresponding template will contain $SUMMARY(statusAreaMap)$
- formatId: identifier referring to the unique identifier of this map in the declarations section.
- timeSeries: reference to an inputVariable (declared in the report or in the declarations section).
- mapId: reference to a valueAttributeMap;

Tables (overview)

Tables are available in different types, see below. The similarity between them is that they are referenced with the same template tag.

In the Table element the variableId's are specified that are used to create a table. The TableFormat controls how the table is formatted, i.e. the row and column information and how the data is displayed in the map.

Related TAG: $TABLE(tableId)$

1. table: *original* table;
2. rowPerLocationHtmlTable: table with a row per location
3. rowPerLocationCSVTable: csv table with a row per locations
4. htmlTable: new style table with same functionality as 'table';
5. countTables:
    a. thresholdsCrossingsTable: table containing numbers (count) of threshold crossings.
    b. thresholdsCrossingsCountTable: new thresholdsCrossingTable
    c. flagCountTable
    d. flagSourceCountTable
6. maximumStatusTable: table containing (coloured) indicating threshold crossings over time.
7. mergedPrecipitationTable: table containing (merged) precipitation figures in specific time intervals
8. systemStatusTable: tables containing system (management) information;
    a. liveSystemStatus: information about live system: MC and FSS('s);
    b. exportStatus: information about exported files/reports;
    c. importStatus: information about files imported;
    d. scheduledWorkflowStatus: information about (next) scheduled workflows;
    e. completedWorkflowStatus: information about number of workflows completed in last 24 hrs.
    f. currentForecastStatus: information about which workflows are set to CURRENT;
    g. logMessageListing list of logmessages (based on a query)
    h. forecastHistory: historic overview of forecasts.

Tables 3 to 7 have references to cascading style sheets.

table

See Table above

rowPerLocationHtmlTable

The rowPerLocationHtmlTable can be configured directly in <report> section, or in the <declarations> section and reffered from the <report> section. This table is a generic table type and contains for each configured location one row with several columns. The data displayed in a particular column are result of the function that is configured for that column.

For each column the following elements can be configured:

- header : text to display in the column header,
- format: format to use for this column. It refers to the styles that are available in the html template file
- width: width of the column
- backgroundColorFunction: function to determine the background color of the cells in the column (see below)
- foregroundColorFunction: function to determine the foreground color of the cells in the column (see below)
- function: function that determines/computes the value displayed in the column (see below)

backgroundColorFunction and foregroundColorFunction

These elements currently support the following functions:

- THRESHOLDCROSSING(FIRST_THRESHOLDCOLOR; variableId)
- THRESHOLDCROSSING(MAX_THRESHOLDCOLOR; variableId)
- INDEXMAXWARNINGLEVELCOLOR(0; variableId)

function

The existing report functions, that are also used in html-templates, can be configured in the columns. For example MAXVALUE(variableId; numberFormat) or LASTVALUETIME(variableId; dateFormat).

Furthermore the following functions are available:

- LOCATIONATTRIBUTE(attribute; <variableId>; <format>)
  Argument attribute refers to the location attributes. The attributes differ per region.
  The other arguments are optional. For example, use argument 'format' if the attribute is a number or a dateTime. VariableId can be omitted : (area;;numberFormat1)
  There are 3 attributes with a fix name: id, name and shortname. These attribute refer always to the location id,name and short name as configured in the locations.xml.

- PARAMETERATTRIBUTE(attribute; variableId; <format>)
  Argument attribute refers to the parameter attributes. The attributes differ per region.
  The argument format is optional. For example, use argument 'format' if the attribute is a number.
  There are 4 attributes with a fix name: id, name,shortname and unit. These attributes refer always to the parameter id,name, short name and unit as configured in the parameters.xml.

- MODULEINSTANCEATTRIBUTE(attribute; moduleInstanceId_or_variableId; <format>)
  Argument attribute refers to the module instance attributes. The attributes differ per region.
  The argument format is optional. For example, use argument 'format' if the attribute is a number.

- THRESHOLDCROSSING(key;variableId;<format>); see note below!
  The argument 'key' specifies which threshold crossing information should be given. Presently the following keys can be used:
  FIRST_THRESHOLDNAME: name of the first level threshold that has been crossed
  FIRST_VALUE: value of the first crossing
  FIRST_DATETIME: date/time of the first crossing
  FIRST_DATE: date of the first crossing
  FIRST_TIME: time of the first crossing
  MAX_THRESHOLDNAME: name of the highest level threshold that has been crossed
  MAX_VALUE: value of the maximum crossing
  MAX_DATETIME date/time of the maximum crossing
  MAX_DATE: date of the maximum crossing
  MAX_TIME: time of the maximum crossing

> ⓘ Please note: If you configure any THRESHOLDCROSSING function in a column, then only the locations with threshold crossing will be printed by default.
> So, if there are no crossings at all, the table is empty. If all locations should always be present in the table, use option onlyWriteRowsWithCrossedThresholds=false

- STATISTICS(statistical function; variableId; numberFormatId)
  The argument 'statistical function' specifies which statistical function should be evaluated. Presently the following functions can be used: COUNT, KURTOSIS, MEAN, MEDIAN, MIN, MAX, RMSQ, RSQUARED, SKEWNESS, STANDARD_DEVIATION, SUM, VARIANCE

Example of the configuration in <report> section

```
<report>

      <locationSetId>AllLocations</locationSetId>

      <rowPerLocationHtmlTable id="tableA" tableStyle="tableStyle3" >

          <column>
              <header>Locatie id</header>
              <format>_data</format>
              <function>LOCATIONATTRIBUTE(id)</function>
          </column>

          <column>
                <header>Locatie name</header>
```

```xml
        <format>_data</format>
        <width>200</width>
          <function>LOCATIONATTRIBUTE(name)</function>
    </column>

    <column>
         <header>Eenheid</header>
         <format>_data</format>
         <function>PARAMETERATTRIBUTE(UNIT;Qobserved)</function>
    </column>


    <column>
            <header>Datum 1e overschrijding</header>
            <format>_data</format>
            <function>THRESHOLDCROSSING(FIRST_DATE;Qobserved;dateFormatThreshold)</function>
    </column>

    <column>
            <header>Tijdstip 1e overschrijding</header>
            <format>_data</format>
            <function>THRESHOLDCROSSING(FIRST_TIME;Qobserved;timeFormatThreshold)</function>
    </column>

    <column>
        <header>Waarde 1e overschrijding</header>
        <format>_data</format>
        <function>THRESHOLDCROSSING(FIRST_VALUE;Qobserved;numberFormatThreshold)</function>
    </column>


    <column>
            <header>Datum maximale overschrijding</header>
            <format>_data</format>
            <function>THRESHOLDCROSSING(MAX_DATE;Qobserved;dateFormatThreshold)</function>
    </column>

    <column>
            <header>Tijdstip maximale overschrijding</header>
            <format>_data</format>
            <function>THRESHOLDCROSSING(MAX_TIME;Qobserved;timeFormatThreshold)</function>
    </column>

    <column>
        <header>Waarde maximale overschrijding</header>
        <format>_data</format>
        <function>THRESHOLDCROSSING(MAX_VALUE;Qobserved;numberFormatThreshold)</function>
    </column>


    </rowPerLocationHtmlTable>

    <template>ReportRowPerLocationTable.html</template>

    <outputFileName>tableA.html</outputFileName>
</report>
```

Example of the configuration in <declarations> section

```
<declarations>
        <rowPerLocationHtmlTableFormat id="rowPerLocationFormat1" tableStyle="tableStyle3" >

          <column>
              <header>Naam</header>
              <format>_data_yellow</format>
            <function>LOCATIONATTRIBUTE(shortname)</function>
        </column>

          <column>
              <header>Tijd maximale overschrijding</header>
              <function>THRESHOLDCROSSING(MAX_DATETIME;Qobserved;dateFormat1)</function>
        </column>


          <column>
              <header>Waarde maximale overschrijding (m3/s)</header>
              <function>THRESHOLDCROSSING(MAX_VALUE;Qobserved;numberFormat1)</function>
        </column>

        <column>
                <header>Tijd maximale waterstand</header>
            <function>MAXTIME(Hobserved;dateFormat1)</function>
        </column>

        <column>
                <header>Maximale waterstand (m)</header>
            <function>MAXVALUE(Hobserved;numberFormat1)</function>
        </column>

      </rowPerLocationHtmlTableFormat>
</declarations>
```

rowPerLocationCSVTable

This table is a generic table type and contains for each configured location one row with several columns, separated by commas. The data displayed in a particular column are result of the function that is configured for that column. For each column the following elements can be configured:

- header : text to display in the column header,
- width: optional, width of the column
- function: function that determines/computes the value displayed in the column.

Note that the xml schema allows element 'format', however the format is not applicable to csv table type Also this report type uses templates. The template should contain at least table name. An example:

{noformat}

$TABLE(myTable)$

{noformat}

The rowPerLocationCsvTable can be configured directly in <report> section, or in the <declarations> section and reffered from the <report> section.

Example of the configuration in <report> section:

```
<report>
        <locationId>M-1000</locationId>
        <locationId>H-2091</locationId>
        <rowPerLocationCsvTable id="myTable">
                <column>
                        <header>Naam</header>
                        <function>LOCATIONATTRIBUTE(shortname)</function>
                </column>
                <column>
                        <header>Tijd maximale overschrijding</header>
                        <function>THRESHOLDCROSSING(MAX_DATETIME;Qobserved;dateFormat1)</function>
                </column>
                <column>
                        <header>Waarde maximale overschrijding (m3/s)</header>
                        <function>THRESHOLDCROSSING(MAX_VALUE;Qobserved;numberFormat1)</function>
                </column>
```

htmlTable

The htmlTable is the successor of the table described earlier. The configuration of this htmlTable is easier and more readable.

Declarations section

In the declarations sections a format of a table needs to be defined. In the figure below, a format of a htmlTable is configured.

Figure 97 Declaration of a htmlTableFormat

The following elements needs to be defined:

- id: unique identifier (as reference to this table);
- tableStyle: a choice of tableStyle which can be influenced by using the corresponding classes in a cascading style sheet. Choices are tableStyle1 to tableStyle10;
- column: indicates what (of one or more Timeseries) should be shown in the columns (options are: location, time, date, locationcritical, parameter, parameters, choice, allSeries);
- row: indicates what (of one or more Timeseries) should be shown in the rows (options are: location, time, date, locationcritical, parameter, parameters, choice, allSeries);
- relativeWholePeriod: definition of a moving period in time to be displayed (in the example above 24 hours before the day containing T0 of the forecast and 24 after the day containing T0 of the forecast (in total 3 days);
- topLeftText: definition of the text to be displayed in the upper left cell of the table;
- cellFormat: format of the cell containing the values;
- topFormat: Format of the column headers: "0,number,0.00" for a number, "0,date,dd/MMM" for a date.
    - In combination with <column>location</column>, the topFormat specifies what should be written in the header of the column: "1" for location name, "2" for parameter name, "3" for parameter unit (N.B. use curly brackets in place of the quotation marks). Example: <topFormat>{1} - {2} ({3})</topFormat>
    - In combination with <column>parameters</column> or <column>allSeries</column>, the topFormat element can be used to specify a custom column header for each data column. For a table with one data column just specify the custom column header in the topFormat element. For a table with multiple data columns, specify multiple custom column headers (one for each column), separated by "#" symbols, in the topFormat element. E.g. "header1#header2#header3". For parameters the headers can also use "0" for the parameter name and "1" for the unit name. The following example shows a configuration for a table with 2 data columns where the parameter name and value are displayed in the header: < topFormat >{0} {1}#{0} {1}</ topFormat >

- To use default headers instead of custom column headers, leave the topFormat element completely empty. If custom column headers are used, then there should be as many custom column headers as data columns in the table, otherwise a warning is logged and default column headers are used instead;
- leftFormat: format of the most left column;
- missingValueText: definition of the missing value character. Choices are: " ", "#", "~~or~~-";
- nullValueText: definition of the null value (NAN) indicator. Choices are: " ", "-", "-999", "no data";
- scrollable: Use this to split the table into two parts, one for the header row(s) and one for the data rows. The data row part refers to the tableStyle for this table with "_scrollable" appended. This can be used to make the data rows scrollable while the header row(s) remain fixed. For this to work the referred style needs to be defined in the report tables .css file.
- tableBackgroundColors: Override background colors in a HTML report for certain hours in the week. Works similar to: 02 Time Series Display Configuration (tableBackgroundColors)

Report Section

In the report itself the reference to a tableFormat and the declaration of the content should take place. The schema prescribes as follows:
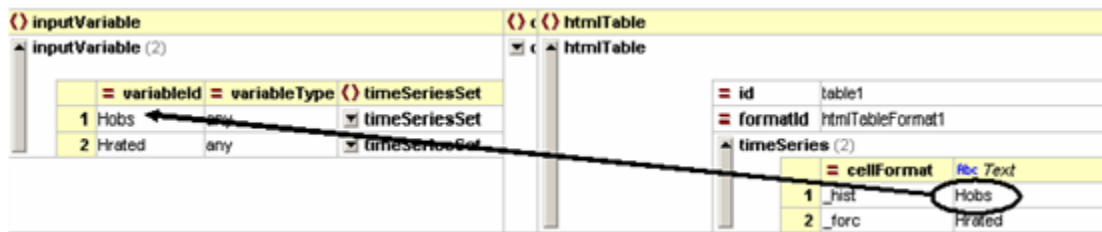
Figure 98 Declaration and reference of a htmlTable in a report

The following elements need to be defined:

- id: identifier to the template tag (in this case: $TABLE(table1)$);
- formatId: reference to the format of this table (to one of the htmlTableFormats in the declarations section);
- timeSeries: reference to an inputVariable (declared in the report or in the declarations section).
- cellFormat: addition to tableStyleX class for adding specific styles to the content of the cell;

Remark: htmlTables can contain more than one timeseries. By adding different cellFormats to the series (see picture above) different styles can be attached for display in the table. In this way you can distinguish the two timeseries in the table!

Detailed explanation

The choice of adding a certain tableStyle to a table supplies you with the opportunity to influence the looks and style of the table, its borders, background and content. By setting the tableStyle a number of style classes are being generated in the html-pages. By declaring these classes in a stylesheet (*.css) and ascribe a certain font family, font size, border color, background color etc. you are able to 'polish' the looks of your reports. If a tableStyle class is not mentioned in a stylesheet, it is being displayed using default settings.

The following classes are generated automatically and are added after the tableStyleX in which X stand for an integer in the range 1 to 10.

class	description	specific for
_beforeT0	date/time indication before time zero (TO) of the forecast	time column (most left column)
_firstAfterT0	date/time indication of the first occurrence after time zero (T0) of the forecast	time column (most left column)
_afterT0	date/time indication after time zero (TO) of the forecast	time column (most left column)
_data	default indication of data content of a cell	data cells
_*anyString*	user defined cellFormat	data cells
_datamax	**addition** to current style if value is maximum of that series (_data_datamax or _anyString_datamax)	data cells
_leftcol	default indication of a row header	
_header	default indication of a column header	
_threshold_n	indication of threshold level (n=0,1,2,...)	threshold tables (colouring of backgrounds)

countTables

thresholdsCrossingsTable

ⓘ   Notice: the thresholdCrossings**Count**Table (see further below) is a newer version of this type of table

A thresholdCrossingsTable is a table in which the number of thresholds for each level are counted. The number given in the table suggests with the 'worst' case situation. When a timeseries crosses a number of thresholds in a forecast, only the 'worst' threshold crossings are counted. An example of a thresholdsCrossingsTable is given below.

Figure 99 Example of a thresholdCrossingTable (NE Region)

Declarations section

In the declaration sections the layout of the table needs to be defined.

Figure 100 Declaration of a thresholdsCrossingsTable

The following elements needs to be defined:

- id: unique identifier (as reference to this table);
- tableStyle: a choice of tableStyle which can be influenced by using the corresponding classes in a cascading style sheet. Choices are tableStyle1 to tableStyle10;
- thresholdsCrossingsCounterHeaderText: String to be displayed in the table header;
- thresholdsCrossingsTableUpCrossingsHeaderText: **not in use anymore: can be left blank.**
- thresholdsCrossingsTableDownCrossingsHeaderText: **not in use anymore: can be left blank.**
- cellFormat: *any string* but "_thresholds" is recommended: style class of the table header in combination with the typeId (range always starts with a 0);
- topFormat; **not in use anymore, can be left blank** ;
- colWidth: width of a column in pixels;
- missingThresholdsText: Choice for missing threshold. Choices are: " ", "#", "~~ or~~ -";
- noThresholdsCrossed: definition of the null value (NAN) indicator. Choices are: " ", "-", "-999", " no data";
- countIndividualThresholds: true if individual threshold crossings must be counted; false (= default) if per location only the worst case must be counted
- thresholdTypes: array of thresholds, assuming a logical order
  o typeId: textual indication of corresponding threshold level;
  o name: indication of the text to appear in column headers;

Report section

A thresholdsCrossingsTable should be defined in the report as well. E.g. this table needs to be 'filled' with its corresponding timeseries.

Figure 101 Declaration of a thresholdsCrossingsTable in a report

The following elements needs to be defined:

- id: identifier to the template tag (in this case: $TABLE(table1)$);
- formatId: reference to the format of this table (to one of the thresholdsCrossingsTable in the declarations section);
- mergeLocations: boolean indicator. True means: treat all locations of mentioned timeseries together for combined assessment. False means: extract individual timeseries so that every row indicates one location (timeseries);
- Choice between
  o table; --> this can be used to display a table 'in' another one.
  o tableRow
- formatId: reference to the thresholdsCrossingsTable format;
- id: identifier (mainly for own use/comment);
- timeSeries:
- mapId: reference to a valueAttributeMap;
- Text: reference to an inputVariable (declared in the report or in the declarations section).

Since this type of table is a table in which you can aggregate data (which means combine timeseries) the following option is available: mergeLocations.

thresholdCrossingCountsTable

A thresholdCrossingsCountsTable displays threshold crossing counts depending on which thresholds have been crossed within a given time period. The thresholdCrossingCountsTable is a new version of the thresholdsCrossingsTable. A thresholdCrossingCountsTable has the same layout as a thresholdCrossingCountsTab in the thresholdOverviewDisplay for consistency.

Declarations section

In the declaration sections the layout of the table needs to be defined in a thresholdCrossingCountsTableFormat.

The following options are available:

- id: identifier of this table format.
- thresholdGroupId: id of a thresholdGroup that is defined in the thresholds configuration file. This table displays all thresholds in the specified thresholdGroup.
- relativePeriod: Relative time period for this table. The time period is relative to timeZero. A relative period can be e.g. -3 to +3 hours or e.g. +3 to +6 hours (relative to timeZero).
- countAllActiveThresholds: If true, then this table counts all thresholds that are active (all thresholds that have been crossed). If false, then for a given location this table only takes into account the active threshold with the most severe warning level. Default is false.
- countWarningAreas: If true, then this table counts warning areas, as follows. It is possible to define warning areas for a levelThresholdValue in the thresholdValueSets configuration file. If a crossed levelThresholdValue has warning areas defined, then the number of warning areas is counted for that levelThresholdValue. If a crossed levelThresholdValue has no warning areas defined, then a count of 1 is used for that levelThresholdValue. If this option is false, then for each crossed levelThresholdValue a count of 1 is used (i.e. the warning areas are ignored). Default is true.
- noThresholdsDefinedText: If specified, then this text is shown in cells that correspond to data for which no thresholds are defined. Default is empty space.
- noDataAvailableText: If specified, then this text is shown in cells for which no data is available. Default is "n/a".
- crossingCountZeroText: If specified, then this text is shown in cells for which there are no threshold crossings. Default is "-".
- tableHeaderText: String to be displayed in the table header. The relative period and thresholdGroup name will be appended to this string.
- columnWidth: The width of the columns in the table.
- tableStyle: The tableStyle to use for this table. The available tableStyles are defined in the report tables .css file.
- cellFormat: The cellFormat to use for this table. The available cellFormats for the configured tableStyle are defined in the report tables .css file.
- scrollableTable: Use this to split the table into two parts, one for the header row(s) and one for the data rows. The data row part refers to the tableStyle for this table with "_scrollable" appended. This can be used to make the data rows scrollable while the header row(s) remain fixed. For this to work the referred style needs to be defined in the report tables .css file.

Report section

In the reports section define a thresholdCrossingCountsTable.

The following options are available:

- id: identifier for the template tag (in this case: $TABLE(table1)$);
- formatId: reference to the format of this table (to one of the thresholdCrossingCountsTableFormats in the declarations section);
- mergeLocations: boolean indicator. True means: treat all locations of mentioned timeseries together for combined assessment. False means: extract individual timeseries so that every row indicates one location (timeseries);
- Choice between
  o timeSeries;
  o table; --> this can be used to display a table within another one.
  o tableRow

flagCountsTable

FlagCountsTable is available since Delft-FEWS release 2011.01. A FlagCountsTable displays flag counts depending on the flags of the values in a time series within a given time period.

Location	Parameter	Reliable	Doubtful	Unreliable	Missing	Total
Cowbeech Weir	Measured Discharge	65	10	10	15	100
Romsey (Botley Road)	Measured Discharge	65	10	10	15	100
Crosslands Drive	Measured Discharge	100	0	0	0	100

Example of a flagCountsTable

Declarations section

In the declaration section the layout of the table needs to be defined in a flagCountsTableFormat.

The following options are available:

- id: identifier of this table format.
- tableStyle: The tableStyle to use for this table. The available tableStyles are defined in the report tables .css file.
- hyperlinkUrl: Optional URL. If specified, then the location name for each time series will be a hyperlink that refers to this URL. It is possible to insert the following tags: !LOCATION_ID!, !LOCATION_NAME!, !PARAMETER_ID! and !PARAMETER_NAME!. The !LOCATION_ID! tag will be replaced with the location id for the time series. The !LOCATION_NAME! tag will be replaced with the location name for the time series. The !PARAMETER_ID! tag will be replaced with the parameter id for the time series. The !PARAMETER_NAME! tag will be replaced with the parameter name for the time series. This can e.g. be used to link each row in this FlagCountsTable to a page (or an anchor within a page) that contains a FlagSourceCountsTable with more detailed information about the time series for that row.
- scrollableTable: Use this to split the table into two parts, one for the header row(s) and one for the data rows. The data row part refers to the tableStyle for this table with "_scrollable" appended. This can be used to make the data rows scrollable while the header row(s) remain fixed. For this to work the referred style needs to be defined in the report tables .css file.

Configuration example:

```
<flagCountsTableFormat id="flagCountsTableFormat">
        <tableStyle>tableStyle1</tableStyle>
        <hyperlinkUrl>report.html#!LOCATION_NAME!_!PARAMETER_NAME!</hyperlinkUrl>
</flagCountsTableFormat>
```

Report section

In the reports section define a flagCountsTable.

The following options are available:

- id: Identifier for this FlagCountsTable that is used in the report template html file in the table tag (e.g: $TABLE(table1)$).
- formatId: The id of the FlagCountsTableFormat to use for this FlagCountsTable.
- inputVariableId: One or more ids of inputVariables that are defined at the start of this report. For each time series in the inputVariable(s), there will be one row in the table with the location, parameter and flag counts for that time series. For a given time series this uses only the data within the relativeViewPeriod that is defined for that time series in the timeSeriesSet. If a timeSeriesSet contains multiple time series (e.g. a locationSet), then for each time series in the timeSeriesSet a separate row is created.

Configuration example:

```
<flagCountsTable id="flagCountsTable" formatId="flagCountsTableFormat">
        <inputVariableId>Cowbeech</inputVariableId>
        <inputVariableId>Romsey</inputVariableId>
        <inputVariableId>CrosslandsDrive</inputVariableId>
</flagCountsTable>
```

flagSourceCountsTable

FlagSourceCountsTable is available since Delft-FEWS release 2011.01. A FlagSourceCountsTable displays counts of flag sources depending on the flag sources of the values in a time series within a given time period. The flag source for a value contains the reason why that value got a certain flag. For example if a value was rejected by a "hard max" validation rule, then it gets flag unreliable and flag source "hard max".

Cowbeech Weir, Measured Discharge					
Validation Rule	Reliable	Doubtful	Unreliable	Missing	Total
Soft Min	0	0	0	0	0
Hard Min	0	0	0	0	0
Soft Max	1	8	0	0	9
Hard Max	0	0	0	0	0
Rate Of Change Rise	0	0	0	0	0
Rate Of Change Fall	0	0	0	0	0
Same Reading	5	0	1	5	11
Temporary Shift	0	0	0	0	0
Series Comparison	0	0	0	0	0
Flags Comparison	0	0	0	0	0
Manual	0	0	0	0	0
Modifier	0	0	0	0	0
None	59	2	9	10	80
Total	65	10	10	15	100

Example of a flagSourceCountsTable

Declarations section

In the declaration section the layout of the table needs to be defined in a flagSourceCountsTableFormat.

The following options are available:

- id: identifier of this table format.
- tableStyle: The tableStyle to use for this table. The available tableStyles are defined in the report tables .css file.
- scrollableTable: Use this to split the table into two parts, one for the header row(s) and one for the data rows. The data row part refers to the tableStyle for this table with "_scrollable" appended. This can be used to make the data rows scrollable while the header row(s) remain fixed. For this to work the referred style needs to be defined in the report tables .css file.

Configuration example:

```
<flagSourceCountsTableFormat id="flagSourceCountsTableFormat">
        <tableStyle>tableStyle1</tableStyle>
</flagSourceCountsTableFormat>
```

Report section

In the reports section define a flagSourceCountsTable.

The following options are available:

- id: Identifier for this FlagSourceCountsTable that is used in the report template html file in the table tag (e.g: $TABLE(table1)$).
- formatId: The id of the FlagSourceCountsTableFormat to use for this FlagSourceCountsTable.
- inputVariableId: The id of an inputVariable that is defined at the start of this report. The time series of this inputVariable is used for this table. This table shows for each validation rule (hard max, hard min, rate of change, etc.) the number of values that were rejected because of that validation rule. This uses only the data within the relativeViewPeriod that is defined for the time series in the timeSeriesSet. If the timeSeriesSet contains multiple time series (e.g. a locationSet), then an error message is given.

Configuration example:

```
<flagSourceCountsTable id="flagSourceCountsTable1" formatId="flagSourceCountsTableFormat">
        <inputVariableId>Cowbeech</inputVariableId>
</flagSourceCountsTable>
```

maximumStatusTable

A maximumStatusTable indicates, by colouring, when certain threshold levels are crossed. In this type of table, the rows should be defined individually and can contain more than one series. The boolean value 'mergLocation' plays an important role in combining the locations or treat them individually.
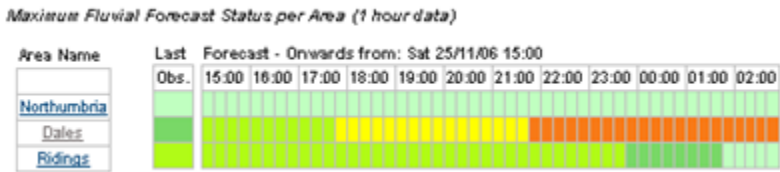
Figure 102 Example of a maximumStatusTable (NE Region)

Declarations section

In the declaration sections the layout of the table needs to be defined.
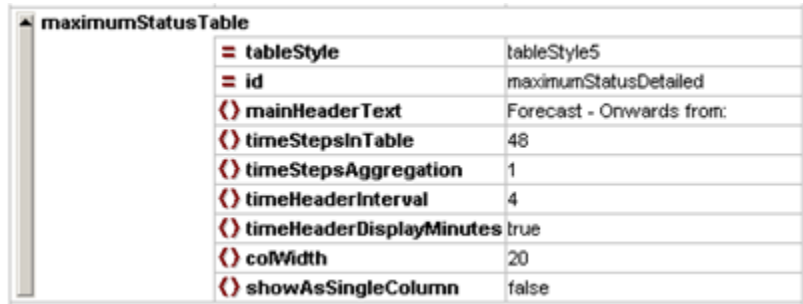
Figure 103 Declaration of maximumStatusTable

The following elements needs to be defined:

- tableStyle: a choice of tableStyle which can be influenced by using the corresponding classes in a cascading style sheet. Choices are tableStyle1 to tableStyle10;
- id: unique identifier (as reference to this table);
- mainHeaderText: a textual string which is displayed in the table header;
- timeStepsInTable: integer number of timesteps to be displayed in the table. This should be derived from the relativeViewPeriod of the corresponding timeSeries to add:
- timeStepsAggregation: integer number of timesteps to be aggregated (=taken together). Worst status is being displayed.
- timeHeaderInterval: integer number for aggregating the headers of the cells or not. Number '1' means 'no aggregation' so every column has got its own header.
- timeHeaderDisplayMinutes: boolean value for having the minutes displayed;
- colWidth: integer value for the width of the cells;
- showAsSingleColumn: boolean value for displaying the timeseries into one column only (true). If set to 'true' the last value of the timeseries is considered.

Reports Section

In the report itself the necessary timeseries needs to be assigned to the table.

Figure 104 Declaration of a maximumStatusTable in a report

The following elements need to be defined:

- id: identifier to the template tag (in this case: $TABLE(table1)$);
- formatId: reference to the format of this table (to one of the maximumStatusTables in the declarations section);
- mergeLocations: boolean indicator. True means: treat all locations of mentioned timeseries together for combined assessment. False means: extract individual timeseries so that every row indicates one location (timeseries);
- table: add individual maximumStatusTables here for a correct visualisation (see detailed explanation: 'two tables into one'). In most cases 2: first for observed values, second for forecast table.

For each table:

- formatId: reference to the format of this table (to one of the maximumStatusTables in the declarations section);
- id: identifier (used for comments only)
- mergeLocations: boolean indicator. True means: treat all locations of mentioned timeseries together for combined assessment. False means: extract individual timeseries so that every row indicates one location (timeseries);
- tableRow: (1 or more)
  o formatId: reference to the thresholdsCrossingsTable format;
  o id: identifier (mainly for own use/comment);
  o timeSeries: (1 or more)
- mapId: reference to a valueAttributeMap;
- Text: reference to an inputVariable (declared in the report or in the declarations section).

ⓘ **RelativeViewPeriod**

The maximumstatusTable was developed with the focus on the status of forecasted timeseries. When using the table to show the status in the past (eg to what extent timeseries are available), it is necessary to place the T0 in the past, at the beginning of the period to be displayed. This is contradictory to normal practice.

Detailed Explanation: Two tables into one

In fact, the maximumStatusTable is designed as visualised below. To create a nicely aligned table the 'two timeseries tables' (the one with the 'observed' values and the one with the forecast series) are put in individual cells of the *outer table.* So the outer table only consist of two cells. The left cell contains the observed table, the right cell contains the forecast table. The outer table itself needs to be declared as well!! The report declaration (see above) can be inspected to see this one in practice.

observed (single) column
forecast columns
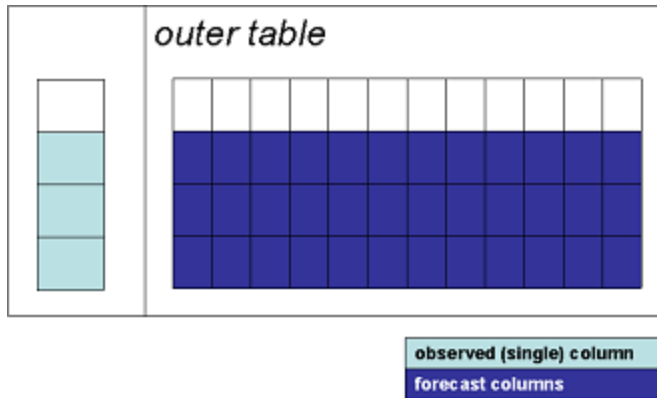
Detailed Explanation: Wide variation of tables

The variation for displaying maximumStatus information is wide. The combination of relativeViewPeriod (length of forecast series), timestep and the desire to aggregate timesteps can all be implemented. The calculation should be correct. If not, several messages will be shown.

Some examples

Source: 12 hrs of 15min data
Display: maximumStatus with 15 minute data, time header by the hour (with minutes)
Configuration

	value	explanation
timeStepsInTable	48	(12*4=48 timesteps)
timeStepsAggregation	1	each column represent 1 timestep
timeHeaderInterval	4	4 columns have a merged header (hour)
timeHeaderDisplayMinutes	true	minute indication

Result: table with 48 time columns with 12 aggregated headers visualising the hour with a minute indication (like in first figure of this section)

Source: 6 hrs of 15min data
Display: maximumStatus with hourly data (no minute indication)
Configuration

	value	explanation
timeStepsInTable	24	(6*4=24 timesteps)
timeStepsAggregation	4	4*15min aggregate to 1 hour
timeHeaderInterval	1	each hour 'column' has its own header
timeHeaderDisplayMinutes	false	no minute indication in header

Result: table with 6 (time) columns indicating the 'worst' status of that hour.

Example thresholdsCrossingsTable/maximumStatusTable

In a configuration the following timeSeries/InputVariables exist:

Name	Parameter	LocationSet	Locations
Catchment1_Waterlevel_Obs	H.obs	Catchment1_Locs	Loc1 Loc2 Loc3
Catchment1_Waterlevel_For	H.simulated	Catchment1_Locs	Loc1 Loc2 Loc3
Catchment2_Waterlevel_Obs	H.obs	Catchment2_Locs	Loc4 Loc5 Loc6
Catchment2_Waterlevel_For	H.simulated	Catchment2_Locs	Loc4 Loc5 Loc6

The geographical hierarchy is that Area 1 contains 2 Catchments (Catchment1 and Catchment2)

The Region overview should be configured that all catchments belonging to that area are 'put' into one row which is describing the status of that area. This is valid for both the observed as the forecast timeseries. The 'mergeLocations' variable should be put to 'true' because all locations should be merged (combined).

The Area overview should be configured in such way that all catchments are in separate rows. This is valid for both the observed as the forecast timeseries. The 'mergeLocations' variable should be put to 'true' because all locations should be merged (combined).

The Catchment overview forms the exception here. With mergeLocations set to 'False', the corresponding locationSet is extracted into the individual locations and so every location has got its own row.

For the two tables for which this is valid, the last example does not give much additional value for a thresholdsCrossingsTable. Then each row (which is equal to one locatation) will have a '1' in one of the cells. A maximumStatusTable supplies more value because it will indicate **when** this (maximum) threshold will be reached.
See below mentioned (simplified) figures.

mergedPrecipitationTable

A mergedPrecipitationTable contains both observed rainfall as well as forecast rainfall, preferably in P.merged timeseries. Data can be visualised in configurable time intervals compared to T0 and will appear in separate columns. Additionally, a *single column* can be added to visualise any parameter (e.g. CWI). An example can be found below (without extra column). In the example below, actually two tables are plotted next to each other. The left table (with names) contains the historical date. The one on the right hand side contains the forecast timeseries and has no name column. A table like this has two header rows to be defined by the user.

Time (hours)	-24 to 0	-12 to 0	-6 to 0	-3 to 0	0 to 3	3 to 6	6 to 12	12 to 24
	(mm)	(mm)	(mm)	(mm)	(mm)	(mm)	(mm)	(mm)
Alston (South Tyne Catchment)	12.7	0.7	0.1	0.1	0	0.2	0	-
Bedburn (Bedburn Beck Catchment)	18.8	3.4	0.2	0.2	0.2	0	0	-
Burn Hall (Browney Catchment)	10.8	2.7	0.1	0	0	0	0	-
Wooler (Till Catchment)	9	0.1	0.1	0.1	0	0.3	1.1	-
Copley (Gaunless Catchment)	18.6	3.9	0.3	0.1	0.3	0.1	0	-
East Gate (Rookhope Burn Catchment)	10.8	0.6	0	0	0.1	0	0	-
Hartburn (Hart Burn Catchment)	7.6	0.8	0.1	0	0	1.2	3.8	-
Kirknewton (Glen Catchment)	7.1	0.2	0.1	0	0	0.6	0.8	-
Middleton Bridge (Wansbeck Catchment)	7.4	0.8	0.1	0	0	1.2	3.8	-
Nunnykirk (Fontburn Catchment)	7.5	0.8	0.2	0	0	1.2	3.8	-
Otterburn Ultrasonic (Rede Catchment)	8.8	0.1	0	0	0	2.2	6.1	-
Shillmoor (Usway Burn Catchment)	7.1	0.3	0.1	0	0	1.3	2.9	-
Stamfordham (Pont Catchment)	6.6	1.8	0	0	0	0.2	0.2	-
Team Valley (Team Catchment)	8	3	0.2	0.1	0	0	0	-
Wearhead (Killhope Burn Catchment)	13.9	0.5	0	0	0	0.1	0	-
Woolsington (Ouseburn Catchment)	7.2	2.3	0	0	0	0	0	-

Figure 105 Example of a mergedPrecipitationTable (NE Region)

Declaration Section

In the declaration section the layout of the table needs to be defined.

Figure 106 mergedPrecipitationTable configuration in the declarations section

The following elements need to be defined.

- id: unique identifier (as reference to this table);
- tableStyle: a choice of tableStyle which can be influenced by using the corresponding classes in a cascading style sheet. Choices are tableStyle1 to tableStyle10;
- headerRow1Text: text to be displayed in table header (first line);
- headerRow2Text: text to be displayed in table header (second line);
- dataColumns: definition of individual column with a time interval for aggregating precipitation. Each column should contain:
  - columnHeader: text to be displayed in column header (first line);
  - unitHeader text to be displayed in column header (second line);
  - relativeViewPeriod: period to aggregate the data;
    - unit: unit to aggregate. Choices: day/hour/minute/second/week
    - start: start of interval (in time unit) compared to T0;
    - end: end of interval (in time unit) compared to T0;
- nameColumnWidth: integer value of width of the name column (most left column (if present));
- dataColumnWidth: integer value of width of the data columns;
- suppressNameColumn boolean value for setting the name column visible or not. This column, when visible, is filled with the (full) location name of the timeseries which is visualised in this table.

Report section

In the report section the content (timeseries) are 'attached' to this table.

Figure 107 mergedPrecipitationTable in the report section.

The mergedPrecipitationTable in the report section is (very) easy to define. The rule of the 'outer table' is valid here as well. To align the historical and the forecast table nicely, the outer table contains both P.merged tables.

The following elements need to be defined:

- id: identifier (reference to the template tag)
- formatId: reference to the format of this table (to one of the mergedPrecipitationTables in the declarations section);
- table:
    - id: identifier (used for comments only)
    - formatId: reference to the format of this table (to one of the mergedPrecipitationTables in the declarations section)
    - timeSeries: reference to an inputVariable.

SystemStatusTables

SystemStatusTables display information about the status and behaviour of the FEWS system itself (like in the System monitor).

SystemStatysTables come in different types:

- liveSystemStatus: information about live system: MC and FSS('s);
- exportStatus: information about exported files/reports;
- importStatus: information about files imported;
- scheduledWorkflowStatus: information about (next) scheduled workflows;
- completedWorkflowStatus: information about number of workflows completed in last 24 hrs.
- currentForecastStatus: information about which workflows are set to CURRENT;
- logMessageListing list of logmessages (based on a query)
- forecastHistory: historic overview of forecasts.

In most tables it is possible to add 'benchmark' data to compare the actual and the desired/required situation. The configuration of such a table requires the definition of this benchmark value. Such a table contains a 'Item', "Benchmark' and a 'Status' column.

Besides a 'benchmark' (something to compare the actual status with) additional fields (columns from the database) can be included in the table. A specific boolean value (showOutputFieldsOnly) can be used to either include or exclude these benchmark columns. In most tables this boolean is set to 'False' because most tables contain both status information as well as additional (meta)information. See figure below.
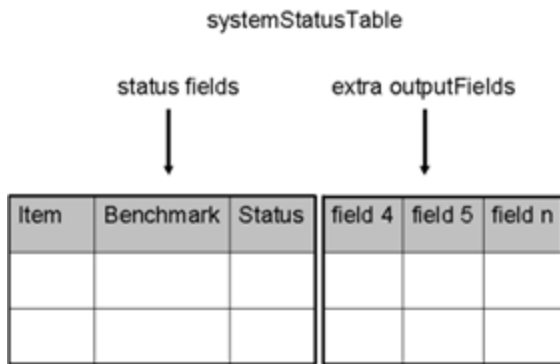
Figure 108 A systemStatusTable is divided into a status part and an extraOutputFields part.

Each type will be briefly explained:

> ⓘ **Global variables**
>
> Notice that for the systemStatus reports the global variable "REPORTS_LEFTCOLMCID" should be included in the global properties. In case of a dual-MC, also "REPORTS_RIGHTCOLMCID" should be included. In a Stand-Alone, use REPORTS_LEFTCOLMCID=none

liveSystemStatus

A liveSystemStatusTable displays information about the status and behaviour of the live system components (MasterController and Forecasting Shell Server(s))

System Status - MC01

Item	Benchmark	Status
MC01	**Alive**	Alive
FSS00	**Alive**	Alive
Size of local datastore	**600 Mb**	728
FSS01	**Alive**	Alive
Size of local datastore	**600 Mb**	726
Log Table Size	**50,000**	16893

Figure 109 Example of a liveSystemStatus table (NE Region)

Declarations Section

Figure 110 Example of the configuration of a liveSystemStatus table in the declarations section

The following elements need to be defined:

- tableStyle: a choice of tableStyle which can be influenced by using the corresponding classes in a cascading style sheet. Choices are tableStyle1 to tableStyle10;
- id: unique identifier (as reference to this table);
- statusTableSubType: Choice for one of the subtypes of systemStatusTables: Choices are: liveSystemStatus, exportStatus, importStatus, scheduledWorkflowStatus, completedWorkflowStatus, currentForecastStatus, logMessageListing, forecastHistory;
- tableTitle: a text for a title for this table;
- headerRows: integer value for the number of header rows;
- itemHeader: header text for the 'Item' column;
- benchmarkHeader: header text for the 'Benchmark' column;
- statusHeader: header text for the 'Status' column;

- statusHeaderSplit: value indicating the number of header rows in the status column.
- showOutputFieldsOnly: boolean value for displaying the outputfields only.

Figure 111 Example of the configuration of a liveSystemStatus table in the report section

The following elements need to be defined:

- id: identifier to the template tag ($TABLE(liveSystemStatusTable)$);
- formatId: reference to the format of this type of systemStatus table in the declarations section);
- mcStatusQuery:
- itemTextStatus: left blank will display MC name;
- benchmarkTextStatus: benchmark text;
- fssStatusQuery:
- itemTextStatus: left blank will display FSS name;
- benchmarkTextStatus: benchmark text for status;
- itemTextSize: text for size query;
- benchmarkTextStatus: benchmark text for size
- singleRecordQuery
- tableName: tablename in local datastore;
- inputfield reference to column in table (to query);
- recordId reference to record in table:
- statusField reference to column in table (to display):
- itemText: text to be displayed (can be empty)
- benchmarkText: benchmark text

(keep in mind that "REPORTS_LEFTCOLMCID" should be included in the global properties (REPORTS_LEFTCOLMCID=none for Stand Alone)

exportStatus table

A exportStatus table displays information about the status of a number export features of the system, such as:
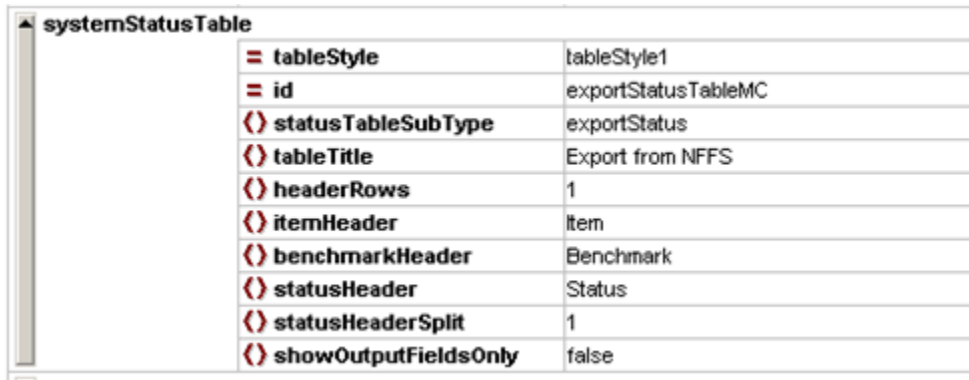
- last occurence of a (export) workflow (by workflowStatusQuery);
- nr of files present in an export directory (by logMessageParseQuery);
- transfer speed of exporting files (by logMessageParseQuery);

Export from NFFS - MC01

Item	Benchmark	Status
Last Export_Current Run (Dispatched)	< 1 hr	27/11/2006 11:16 GMT
Last Report_Export Run (Dispatched)	< 1 hr	not known
Nr. of files in Export to RTS folder	0	68
Average transfer speed of Report export	> 1000 Kb/s	254.02
Nr of current forecast reports exported to webserver	246	4
Last Archive Run (Dispatched)	< 1 day	27/11/2006 04:00 GMT
Nr. of files in export archive folder	0	165

Figure 112 Example of an exportStatus table (NE Region)

Declarations Section

Figure 113 Example of the configuration of an exportStatus table in the declarations section

The following elements need to be defined:

- tableStyle: a choice of tableStyle which can be influenced by using the corresponding classes in a cascading style sheet. Choices are tableStyle1 to tableStyle10;
- id: unique identifier (as reference to this table);
- statusTableSubType: Choice for one of the subtypes of systemStatusTables: Choices are: liveSystemStatus, exportStatus, importStatus, scheduledWorkflowStatus, completedWorkflowStatus, currentForecastStatus, logMessageListing, forecastHistory;
- tableTitle: a text for a title for this table;
- headerRows: integer value for the number of header rows;
- itemHeader: header text for the 'Item' column;
- benchmarkHeader: header text for the 'Benchmark' column;
- statusHeader: header text for the 'Status' column;
- statusHeaderSplit: value indicating the number of header rows in the status column.
- showOutputFieldsOnly: boolean value for displaying the outputfields only.

Report Section

Figure 114 Example of the configuration of an exportStatus table in the report section

This table is constructed using two types of queries:

- workflowStatusQuery;
- logMessageParseQuery

The following elements need to be defined:

- id: identifier to the template tag ($TABLE(liveSystemStatusTable)$);
- formatId: reference to the format of this type of systemStatus table in the declarations section);
    - workflowStatusQuery:
        - workflowId:
        - itemText: text in 'Item' column;
        - benchmarkTextStatus: text in the 'Benchmark' column;
        - selectCompletedWorkflowsOnly: boolean indicating completed or scheduled workflows;
        - statusField: field from the taskRunsCompletions table indicating whether a workflow has been completed or not;
    - logMessageParseQuery
        - itemText: text in 'Item' column;
        - benchmarkTextStatus: text in the 'Benchmark' column;
        - logEntryEventCode: eventcode to filter on;
        - keyword: typical/unique keyword to parse for;
        - displayWord: integer value for the word to display (n-th word in this log message);

(keep in mind that "REPORTS_LEFTCOLMCID" should be included in the global properties (REPORTS_LEFTCOLMCID=none for Stand Alone)

importStatus table

An importStatus table displays information about the datafeeds which have been imported, how many of them were read and how many failed to be imported. The frequency of the files imported can be (visually) compared with a benchmark figure. See below for an example.

Import into NFFS - MC01

Data feed	Benchmark (time from now)	Last import time	Last file imported	Nr. of files read	Nr. of files failed
Astronomical	<330 days	07/11/2006 11:58 GMT	TEES.xml	3	0
MOSurgeCS3	<8 hrs	27/11/2006 08:57 GMT	MOEAEUSURGE20061127060000.DAT	1	0
MOWaveForecast	n/a	27/11/2006 09:58 GMT	MOEAUKWAVE20061127060000.DAT	1	0
MOWindForecast	<8 hrs	27/11/2006 09:58 GMT	MOEAWIND20061127060000.DAT	1	0
NWPForecast	<8 hrs	27/11/2006 10:38 GMT	EAHYNEFS20061127103520964.xml	2	0
NWPTempForecast	<8 hrs	27/11/2006 10:38 GMT	EAHYNEFS20061127103520416.xml	2	0
RTS	<1 hr	27/11/2006 11:07 GMT	NETSNEFS20061127110625188.xml	2	0
RadarActual	<1 hr	27/11/2006 11:07 GMT	EAHYNEFS20061127110039894.xml	1	0
RadarForecast	<1 hr	27/11/2006 10:58 GMT	EAHYNEFS20061127105533269.xml	2	0
Surge	<12 hrs	27/11/2006 03:37 GMT	MET20061127030729A.xml	1	0
SynopticForecast	<8 hrs	18/10/2006 05:17 GMT	NEWEAT20061018050640A.xml	1	0
TidalMonitoring	n/a	27/11/2006 11:07 GMT	COMBINEDBODD271106105301.xml	1	0
WindWave	<12 hrs	27/11/2006 03:48 GMT	MET20061127030931A.xml	1	0

Figure 115 Example of an importStatus table (NE Region)

Declarations Section

▲ systemStatusTable

= tableStyle		tableStyle1
= id		importStatusTableMC
{} statusTableSubType		importStatus
{} tableTitle		Import into NFFS
{} headerRows		1
{} itemHeader		Data feed
{} benchmarkHeader		Benchmark
(time from now)
{} statusHeader		Last import time
{} statusHeaderSplit		1
▲ extraOutputFieldHeader		

	Text
1	Last file imported
2	Nr. of files read
3	Nr. of files failed

{} showOutputFieldsOnly	false

Figure 116 Example of the configuration of an importStatus table in the declarations section

The following elements need to be defined:

- tableStyle: a choice of tableStyle which can be influenced by using the corresponding classes in a cascading style sheet. Choices are tableStyle1 to tableStyle10;
- id: unique identifier (as reference to this table);
- statusTableSubType: Choice for one of the subtypes of systemStatusTables: Choices are: liveSystemStatus, exportStatus, importStatus, scheduledWorkflowStatus, completedWorkflowStatus, currentForecastStatus, logMessageListing, forecastHistory;
- tableTitle: a text for a title for this table;
- headerRows: integer value for the number of header rows;
- itemHeader: header text for the 'Item' column;
- benchmarkHeader: header text for the 'Benchmark' column;
- statusHeader: header text for the 'Status' column;
- statusHeaderSplit: value indicating the number of header rows in the status column.
- extraOutputFieldHeader: Additional Field definition specifically for import related topics. Recommended fields are:
  - Last file imported
  - Nr. of files read
  - Nr. of files failed
- showOutputFieldsOnly: boolean value for displaying the outputfields only.

Remark: when defining extraOutputFieldHeaders it is important to maintain the same order in the declarations sections (definition of the fields) and in the report section (referencing the content) otherwise the header and the content will not correspond.

Figure 117 Example of the configuration of an importStatus table in the report section

The following elements need to be defined:

- id: identifier to the template tag ($TABLE(importStatusTable)$);
- formatId: reference to the format of this type of systemStatus table in the declarations section);
- datafeedStatusQuery
  - BenchmarkText
    - id: textual string contain the name of the datafeed
    - text: textual string indicating the benchmark for this datafeed
  - statusField: textual reference to main field in the database table ImportStatus, lastImportTime
- extraOutputField
  - Text: textual references to the fields in the database table ImportStatus to fill the defined ExtraOutputFields (Declaration section). In this case it is recommended to add here:
    - lastFileImported
    - filesReadCount
    - filesFailedCount

(keep in mind that "REPORTS_LEFTCOLMCID" should be included in the global properties (REPORTS_LEFTCOLMCID=none for Stand Alone)

scheduledWorkflowStatus table

A scheduledWorkflowStatus table displays the workflows which are scheduled together with their repeat time and next due time. The figure below illustrates this.

Figure 118 Example of a scheduledworkflowStatus table (NE Region)

Declarations Section

Figure 119 Example of the configuration of a completedWorkflowStatus table in the declarations section

The following elements need to be defined:

- tableStyle: a choice of tableStyle which can be influenced by using the corresponding classes in a cascading style sheet. Choices are tableStyle1 to tableStyle10;
- id: unique identifier (as reference to this table);
- statusTableSubType: Choice for one of the subtypes of systemStatusTables: Choices are: liveSystemStatus, exportStatus, importStatus, scheduledWorkflowStatus, completedWorkflowStatus, currentForecastStatus, logMessageListing, forecastHistory;
- tableTitle: a text for a title for this table;
- headerRows: integer value for the number of header rows;
- itemHeader: header text for the 'Item' column;
- benchmarkHeader: header text for the 'Benchmark' column;
- statusHeader: header text for the 'Status' column;
- statusHeaderSplit: value indicating the number of header rows in the status column.
- extraOutputFieldHeader: Additional Field(s) definition specifically for scheduled workflow related topics. Recommended fields are:
  - Workflows
  - Description
  - MC Id
  - Repeat Time
  - Next Due Time
- showOutputFieldsOnly: boolean value for displaying the outputfields only.

Since this table is not referring to a benchmark (it is just reading the configuration) the value for showOutputFieldsOnly is set to true. Only these fields are displayed.

remark: One reference to an existing workflow is sufficient to extract all scheduled workflows out of the database, that's why it seems that there is only one table row configured here. In fact, this table will be filled with ALL scheduled workflows when configured as above.

Figure 120 Example of the configuration of a scheduledWorkflowStatus table in the report section

The following elements need to be defined:

- id: identifier to the template tag ($TABLE(scheduledWorkflowTable)$);
- formatId: reference to the format of this type of systemStatus table in the declarations section);
- workflowStatusQuery
  - workflowId: textual reference (case-sensitive!) to an existing workflow in the configuration
  - itemText: text in the 'Item' Column (can be left blank in case of this statusTableSubType)
  - benchmarkText: text in the 'Benchmark' column (can be left blank in case of this statusTableSubType)
  - selectCompletedWorkflowsOnly: boolean variable: 'true' refers to query completed workflows (see next table subtype) and 'false' refers to query scheduled workflows (this type)
  - statusField: textual reference to the field in the database table Tasks. In this case taskStatus. In case of *scheduled workflows* the column 'taskRepeatTime' contains a integer value indicating that it is a repeating activity (workflow)
  - extraOutputField
    - Text: textual references to the fields in the database table ImportStatus to fill the defined ExtraOutputFields (Declaration section). In this case it is recommended to add here:
      - workflowId
      - description
      - ownerMcId
      - taskRepeatTime
      - taskPendingSinceTime

(keep in mind that "REPORTS_LEFTCOLMCID" should be included in the global properties (REPORTS_LEFTCOLMCID=none for Stand Alone)

completedWorkflowStatus table

A completedWorkflowStatus table contains an overview of all workflows carried out in the last 24 hours. An example is given below.

Number of runs per workflow (past 24 hrs) - MC01

Workflows	Nr of Runs	Status	
		Nr. of Runs	Nr. Failed
Scheduled Riverflow forecast for complete region	4	4	0
Scheduled Fluvial Fast responding catchments	12	12	0
Scheduled Riverflow Historical	1	1	0
Scheduled Coastal Forecast	12	12	0
Triton Forecast	4	4	0
RollingBarrel_FSS00	1	0	0
RollingBarrel_FSS01	1	0	0

Figure 121 Example of a completedworkflowStatus table (NE Region)

Figure 122 Example of the configuration of a completedWorkflowStatus table in the declarations section

The following elements need to be defined:

- tableStyle: a choice of tableStyle which can be influenced by using the corresponding classes in a cascading style sheet. Choices are tableStyle1 to tableStyle10;
- id: unique identifier (as reference to this table);
- statusTableSubType: Choice for one of the subtypes of systemStatusTables: Choices are: liveSystemStatus, exportStatus, importStatus, scheduledWorkflowStatus, completedWorkflowStatus, currentForecastStatus, logMessageListing, forecastHistory;
- tableTitle: a text for a title for this table;
- headerRows: integer value for the number of header rows;
- itemHeader: header text for the 'Item' column;
- benchmarkHeader: header text for the 'Benchmark' column;
- statusHeader: header text for the 'Status' column;
- statusHeaderSplit: value indicating the number of header rows in the status column.
- statusSubHeader: Additional Field(s) definition specifically for completed workflow related topics. Recommended fields are:
    - Nr. of Runs
    - Nr. Failed
- showOutputFieldsOnly: boolean value for displaying the outputfields only.

Report Section

Figure 123 Example of the configuration of a completedWorkflowStatus table in the report section

The following elements need to be defined:

- id: identifier to the template tag ($TABLE(scheduledWorkflowTable)$);
- formatId: reference to the format of this type of systemStatus table in the declarations section);
- workflowStatusQuery
    - workflowId: textual reference (case-sensitive!) to an existing workflow in the configuration
    - itemText: text in the 'Item' Column (can be left blank in case of this statusTableSubType)
    - benchmarkText: text in the 'Benchmark' column (can be left blank in case of this statusTableSubType)
    - selectCompletedWorkflowsOnly: boolean variable: 'true' refers to query completed workflows (this type) and 'false' refers to query scheduled workflows (see previous table subtype)
    - statusField: textual reference to the field in the database table TaskRuns. In this case taskRunStatus.

(keep in mind that "REPORTS_LEFTCOLMCID" should be included in the global properties (REPORTS_LEFTCOLMCID=none for Stand Alone)

currentForecastStatus table

The currentForecastStatus table gives an overview of which workflows are set to CURRENT. These mentioned workflows in this tables are the same as the marked with a green icon the System Monitor of the Operator Client. An example of this table is given below.

Workflow	Benchmark Dispatch Time	Dispatch Time	T0	What-if Scenario	Description	FDO
RiverFlow_Forecast	<6 hrs	27/11/2006 07:21 GMT	27/11/2006 07:15 GMT	-	Scheduled Riverflow forecast for complete region	
Fluvial_FastResponse_Forecast	<2 hrs	27/11/2006 11:15 GMT	27/11/2006 11:15 GMT	-	Scheduled Fluvial Fast responding catchments	
RiverFlow_Historical	<24 hrs	27/11/2006 07:00 GMT	27/11/2006 00:00 GMT	-	Daily States Run	
CoastalForecast_workflow	<2 hrs	27/11/2006 09:20 GMT	27/11/2006 09:00 GMT	-	Auto Scheduled Forecast	
Triton_Forecast	<2 hrs	27/11/2006 10:30 GMT	27/11/2006 06:00 GMT	-	TRITON scheduled forecast	

Figure 124 Example of a currentForecastStatus table (NE Region)

Declarations Section

▲ **systemStatusTable**		
= **tableStyle**	tableStyle1	
= **id**	currentForecastsMC	
{} **statusTableSubType**	currentForecastStatus	
{} **tableTitle**	Current Forecast	
{} **headerRows**	1	
{} **itemHeader**	Workflow	
{} **benchmarkHeader**	Benchmark
Dispatch Time	
{} **statusHeader**	Dispatch Time	
{} **statusHeaderSplit**	1	
▲ **extraOutputFieldHeader**		
		Text
	1	T0
	2	What-if Scenario
	3	Description
	4	FDO
{} **showOutputFieldsOnly**	false	

Figure 125 Example of the configuration of a currentForecast table in the declarations section

The following elements need to be defined:

- tableStyle: a choice of tableStyle which can be influenced by using the corresponding classes in a cascading style sheet. Choices are tableStyle1 to tableStyle10;
- id: unique identifier (as reference to this table);
- statusTableSubType: Choice for one of the subtypes of systemStatusTables: Choices are: liveSystemStatus, exportStatus, importStatus, scheduledWorkflowStatus, completedWorkflowStatus, currentForecastStatus, logMessageListing, forecastHistory;
- tableTitle: a text for a title for this table;
- headerRows: integer value for the number of header rows;
- itemHeader: header text for the 'Item' column;
- benchmarkHeader: header text for the 'Benchmark' column;
- statusHeader: header text for the 'Status' column;
- statusHeaderSplit: value indicating the number of header rows in the status column.
- extraOutputFieldHeader: Additional Field(s) definition specifically for scheduled workflow related topics. Recommended fields are:
  - T0
  - What-if Scenario
  - Description
  - FDO
- showOutputFieldsOnly: boolean value for displaying the outputfields only.

Report Section

Figure 126 Example of the configuration of a currentForecastStatus table in the report section

The following elements need to be defined:

- id: identifier to the template tag ($TABLE(scheduledWorkflowTable)$);
- formatId: reference to the format of this type of systemStatus table in the declarations section);
- currentForecastQuery
    - workflowId: textual reference (case-sensitive!) to an existing workflow in the configuration
    - itemText: text in the 'Item' Column
    - benchmarkText: text in the 'Benchmark' column
    - statusField: textual reference: **should be "dispatchTime"!**
        - extraOutputField: textual references to specific TaskRun details (see figure above):
            - T0
            - whatIfId
            - description
            - FDO

(keep in mind that "REPORTS_LEFTCOLMCID" should be included in the global properties (REPORTS_LEFTCOLMCID=none for Stand Alone)

logMessageListing table

A logMessageListing table contains logmessages which are available in the Log Browser tab in the System Monitor of the Operator Client. Log messages of a specific type can be queried. By making use of a correct reference to the cascading style sheet this table can be set to 'scrollable' An example of such a table is given in the figure below.

Figure 127 Example of a logMessageListing table (NE Region)

Declarations Section

Figure 128 Example of the configuration of a logMessageListing table in the declarations section

The following elements need to be defined:

- tableStyle: a choice of tableStyle which can be influenced by using the corresponding classes in a cascading style sheet. Choices are tableStyle1 to tableStyle10;
- id: unique identifier (as reference to this table);
- statusTableSubType: Choice for one of the subtypes of systemStatusTables: Choices are: liveSystemStatus, exportStatus, importStatus, scheduledWorkflowStatus, completedWorkflowStatus, currentForecastStatus, logMessageListing, forecastHistory;
- tableTitle: a text for a title for this table;
- headerRows: integer value for the number of header rows;
- itemHeader: header text for the 'Item' column;
- benchmarkHeader: header text for the 'Benchmark' column;
- statusHeader: header text for the 'Status' column;
- statusHeaderSplit: value indicating the number of header rows in the status column.
- extraOutputFieldHeader: Additional Field(s) definition specifically for scheduled workflow related topics. Recommended fields are:
  - Log Creation Time
  - Log Message
  - TaskrunId
- showOutputFieldsOnly: boolean value for displaying the outputfields only. This tableType requires a 'true' here.

Report Section

Figure 129 Example of the configuration of a currentForecastStatus table in the report section

The following elements need to be defined:

- id: identifier to the template tag ($TABLE(scheduledWorkflowTable)$);
- formatId: reference to the format of this type of systemStatus table in the declarations section);
- logMessageQuery
  - logLevelFilter: textual (case-sensitive) reference to one of the log message levels. Choices are: INFO, WARN, ERROR, FATAL.
  - logEntryEventCode: textual (case-sensitive) reference to a specific type of log message. The eventCode is a 'filter' to retrieve certain types or error messages. In this case the "**TaksRun.Completed** " eventCode has been used.
  - statusField: textual reference to the correct field in the LogEntries table in the database (="logLevel");
  - extraOutputField: textual reference to other fields in the LogEntries table which fill the corresponding columns. In this case 3 additional columns need to be filled with information:
    - logCreationTime (creation time of message)
    - logMessage (content of the log message itself)

- taskRunId (reference to the taskrun that throwed this message)

(keep in mind that "REPORTS_LEFTCOLMCID" should be included in the global properties (REPORTS_LEFTCOLMCID=none for Stand Alone)

forecastHistory table

A forecastHistory table provides an overview of all most recent forecasts carried out. The number of forecasts to include is configurable. An example of such a table is given below.

Figure 130 Example of a forecastHistory table (NE Region)

Declarations Section

Figure 131 Example of the configuration of a forecastHistory table in the declarations section

The following elements need to be defined:

- tableStyle: a choice of tableStyle which can be influenced by using the corresponding classes in a cascading style sheet. Choices are tableStyle1 to tableStyle10;
- id: unique identifier (as reference to this table);
- statusTableSubType: Choice for one of the subtypes of systemStatusTables: Choices are: liveSystemStatus, exportStatus, importStatus, scheduledWorkflowStatus, completedWorkflowStatus, currentForecastStatus, logMessageListing, forecastHistory;

- tableTitle: a text for a title for this table;
- headerRows: integer value for the number of header rows;
- itemHeader: header text for the 'Item' column;
- benchmarkHeader: header text for the 'Benchmark' column;
- statusHeader: header text for the 'Status' column;
- statusHeaderSplit: value indicating the number of header rows in the status column.
- extraOutputFieldHeader: Additional Field(s) definition specifically for scheduled workflow related topics. Recommended fields are:
  - Dispatch Time
  - Completion Time
  - T0
  - Workflow
  - What-if Scenario
  - Description
  - FDO
- showOutputFieldsOnly: boolean value for displaying the outputfields only. This tableType requires a 'true' here.
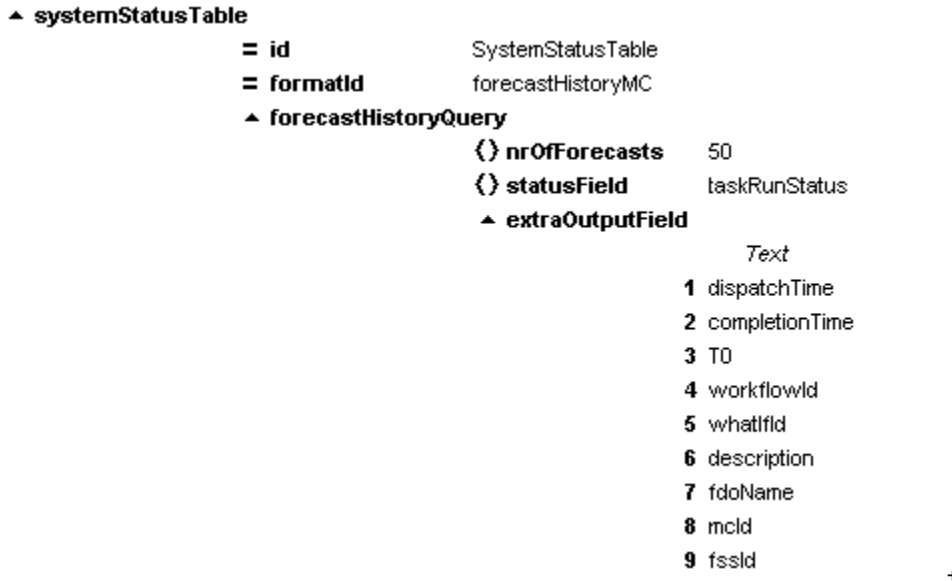
Figure 132 Example of the configuration of a forecastHistory table in the report section

The following elements need to be defined:

- id: identifier to the template tag ($TABLE(scheduledWorkflowTable)$);
- formatId: reference to the format of this type of systemStatus table in the declarations section);
- forecastHistoryQuery
  - nrOfForecasts: integer value referring to the number of most recent forecast to include in this table;
  - statusField: textual reference to the field in the TaskRuns table of the database
  - extraOutputFields: textual references to required fields
    - dispatchTime
    - completionTime
    - T0
    - workflowId
    - whatIfId
    - description
    - fdoName
    - mcId
    - fssId

(keep in mind that "REPORTS_LEFTCOLMCID" should be included in the global properties (REPORTS_LEFTCOLMCID=none for Stand Alone)

forecastThresholdCrossingXml

The forecastThresholdCrossingXml report generates an xml-file containing threshold crossing information for a given ensemble time series. An example of a generated xml file can be found here.

Configuring the forecastThresholdCrossingXml report requires at least two input variables to be configured. One input variable containing the forecast ensemble time series for which the report must be generated, and a second input variable containing the a time series which has the current level (value) for the same locations. Optionally, additional forecast scenario input variables can be defined, each containing forecast time series (no ensemble) for the same locations as the ensemble input variable.

Example config

The example generated xml file was generated using the following configuration:

Example config for forecastThresholdCrossingXml

```xml
<report>
   <inputVariable variableId="forecast_series">
      ...
   </inputVariable>
   <inputVariable variableId="ensemble_series">
      ...
   </inputVariable>
   <inputVariable variableId="level_series">
      ...
   </inputVariable>
   <forecastThresholdCrossingXml fileName="threshold-crossing.xml" locationNameAttribute="REPORT_NAME"
externalLocationIdAttribute="EXTERNAL_ID">
      <forecastThresholdCrossingXmlTemplate>ForecastThresholdCrossing_template.xml<
/forecastThresholdCrossingXmlTemplate>
      <forecastEnsembleTimeSeries>ensemble_series</forecastEnsembleTimeSeries>
      <forecastScenarios>
         <forecastTimeSeries id="1" name="Forecast Policy" numberFormatId="numberFormatId">forecast_series<
/forecastTimeSeries>
      </forecastScenarios>
      <exceedance>
         <numberFormatId>numberFormatId</numberFormatId>
         <minEnsembleMembers>5</minEnsembleMembers>
         <belowMinimumLabel>&lt;10%</belowMinimumLabel>
         <earliestExceedanceTimeStep unit="hour" multiplier="1"/>
         <scenarios>
            <earliest name="earliest"/>
            <percentile name="25%">25</percentile>
            <percentile name="50% (most likely)">50</percentile>
            <percentile name="75%">75</percentile>
            <latest name="latest"/>
         </scenarios>
      </exceedance>
      <currentLevelTimeSeries numberFormatId="numberFormatId">level_series</currentLevelTimeSeries>
      <floodClasses>
         <default>less-than-minor</default>
         <floodClass thresholdId="levelThreshold1">minor</floodClass>
         <floodClass thresholdId="levelThreshold2">moderate</floodClass>
         <floodClass thresholdId="levelThreshold3">major</floodClass>
      </floodClasses>
      <additionalThresholds>
         <thresholdId>levelThreshold3</thresholdId>
         <thresholdId>levelThreshold4</thresholdId>
         <thresholdId>levelThreshold4</thresholdId>
      </additionalThresholds>
      <siteFigures>
         <siteFigure>
            <id>1</id>
            <name>@NAME@Forecast1</name>
            <fileName>@NAME@-forecast-1.png</fileName>
            <type>png</type>
            <alternativeText>Graphical forecast for @NAME@</alternativeText>
            <viewPort left="0" top="0" width="800" height="600"/>
         </siteFigure>
         <siteFigure>
            <id>2</id>
            <name>@NAME@Forecast2</name>
            <fileName>@NAME@-forecast-2.png</fileName>
            <type>png</type>
            <alternativeText>Graphical forecast for @NAME@</alternativeText>
            <viewPort left="100" top="100" width="900" height="700"/>
         </siteFigure>
      </siteFigures>
   </forecastThresholdCrossingXml>
   <template>ForecastThresholdCrossing_template.xml</template> <!-- irrelevant, the report module will use the
<forecastThresholdCrossingXmlTemplate> element to determine the template -->
   <outputSubDir>thresholdcrossing</outputSubDir>
   <outputFileName>crossingTest.html</outputFileName> <!-- irrelevant, the report module will use the
```

```
"filename" attribute of the <forecastThresholdCrossingXml> element to determine the output file name -->
</report>
```

The ForecastThresholdCrossing_template.xml can contain normal report tags which will be replaced as usual. This report type also allows several report specific tags:

- %GENERATION_TIME% : The system time at the moment the report is generated.
- %FORECAST_START_TIME% : the t0 with which the report is generated.
- %NEXT_GENERATION_TIME% : if the workflow in which the report is generated is a scheduled task, the next time at which this task will run is reported. Otherwise: "N/A".
- %PREVIOUS_GENERATION_TIME% : if the workflow in which the report is generated has been run before, the last time at which the workflow was run is reported. Otherwise: "N/A"
- %ENSEMBLE% : this tag will be replaced with the <ensemble> element and all of its sub-elements as described below.

The forecastThresholdCrossingXml report type generates an <ensemble> element for the given ensemble input time series. This ensemble element will first contain a number of <scenario> elements equal to the number of forecast scenario time series specified in the configuration. The id and name attributes of each of these <scenario> elements will match the configured id and name for the <forecastTimeSeries> in the configuration. After the <scenario> elements, there will be a <sites> element in which for each location in the given ensemble time series, the current level (taken from the level time series), figures (determined using the <siteFigures> in the configuration) and threshold crossings for each of the scenarios (configured in the <exceedance> element) will be reported. The thresholds which are to be included in the report are all the thresholds named in the <floodClasses> element as well as the thresholds configured in the <additionalThresholds> element.

With each value reported for the forecast scenario time series as well as the level time series, the report will also contain a "flood-class". This flood class is determined using the <floodClasses> configured. The flood class corresponding to a value is the highest of the configured flood class thresholds which is exceeded by the value, or the configured default flood class if none of the thresholds is exceeded.

For each of the sites, the number of ensemble members which cross a threshold is calculated. This is reported as the "exceedance" in the generated report: the percentage of ensemble members which crossed the threshold. This percentage will be formatted using the number format reference by <numberFormatId> if configured. If at least one but less than the configured <minEnsembleMembers> cross the threshold, and there is a <belowMinimumLabel> configured, than the <belowMinimumLabel> will be reported as the "exceedance" instead. In this example "<10%".

If more than the configured <minEnsembleMembers> cross the threshold, the earliest exceedance times for each of the configured <scenario> elements is reported. To obtain these times, only the ensemble members that cross the threshold are taken into account. The times at which these members first cross the threshold are determined and ordered from earliest to latest. For an <earliest> scenario the first time at which an ensemble member crosses the threshold is reported. For a <latest> scenario, the last time is reported. For the configured <percentile> scenarios, the x-th percentile is taken from the ordered list of crossing times. If the percentile would "fall between two values in the list" their weighted average is taken. If an <earliestExceedanceTimeStep> is configured, each of the reported earliest exceedance times is changed to the closest time in the configured time step.

floodScenarioXml

WORK-IN-PROGRESS (TO DO RUDIE: **FEWS-19742** - Getting issue details... **STATUS**)

AnimatedGIF

An animated GIF is an animation of a grid timeserie in a GIF image format. The advantage of this format is that it can easily be added to HTML pages, without the need to embed extra controls (like with AVI).

Reportsection

The following elements need to be defined:

- id: identifier to the template tag ($ANIMATED_GIF(GIFFILE.gif)$);
- name (optional, at this moment it is not possible to define the filename)

- timeSeriesSet: Definition of the timeserieset to be animated
- movieFrameDurationMillis: defines the speed of the animation
- barLegend: from 2012.02 it is possible to display a bar legend in the animation
- classBreaks: see Grid Display
- geomap: see Grid Display
- width: the width of the image in pixels
- height: the height of the image in pixels

To export a report see the reportExport module instance. However, if you simply want to include your created gif in a html report, you can use <IMG SRC=" gif0_0.gif"> in the report template file, where gif0_0.gif refers to the name of your created gif.

Schematic Status Display Reports

Since 2013_01 it is possible to create reports of Schematic Status Display panels in the form of animations in the formats Avi and animated gif as well as snapshots in the form of Png files and Svg files.
For this purpose there will probably already be some Schematic Status Display configuration in the DisplayConfigFiles of the configuration. In order to create a report of a Schematic Status Display panel, most elements of this configuration can be copied manually to the reports configuration file. For configuring a Schematic Status Display panel please refer to Schematic Status Display configuration.

An example Schematic Status Display that contains the following configuration,

```
<dateFormat id="DF1">
        <timeZone>
                <timeZoneName>CET</timeZoneName>
        </timeZone>
        <dateTimePattern>dd-MM-yyyy</dateTimePattern>
</dateFormat>
<numberFormat id="NF1">
        <pattern>{0,number,##0.00}</pattern>
</numberFormat>
```

must be transferred to the report declarations. The definitions of the date format and number format cannot be copied directly to the report configuration since they have a slightly different format.

```
<reports xmlns="http://www.wldelft.nl/fews" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:
schemaLocation="http://www.wldelft.nl/fews https://fewsdocs.deltares.nl/schemas/version1.0/reports.xsd"
version="1.0">
<declarations>
        <defineGlobal id="ForecastingRegion">IWP-Twentekanalen</defineGlobal>
        <dateFormat id="DF1">
                <timeZone>
                        <timeZoneName>CET</timeZoneName>
                </timeZone>
                <pattern>dd-MM-yyyy</pattern>
        </dateFormat>
        <numberFormat id="NF1">{0,number,##0.00}</numberFormat>
                ...
</declarations>
<report>
...
// Add Ssd report elements here for generating Avi, Gif, Png or Svg.
...
</report>
</reports>
```

When first configuring a Schematic Status Display panel report, the *scadaPanel* section of the Schematic Status Display configuration can be copied directly to the Schematic Status Display Reports configuration. For Svg output the interactive sections will not be supported, so it is recommended to remove these.

Animation in Avi format

With this function an animation of the configured period can be generated for a single Schematic Status Display panel. Use id $SSD_AVI(myId)$ for including the avi file in the reports.

schematicStatusDisplayPanelAvi
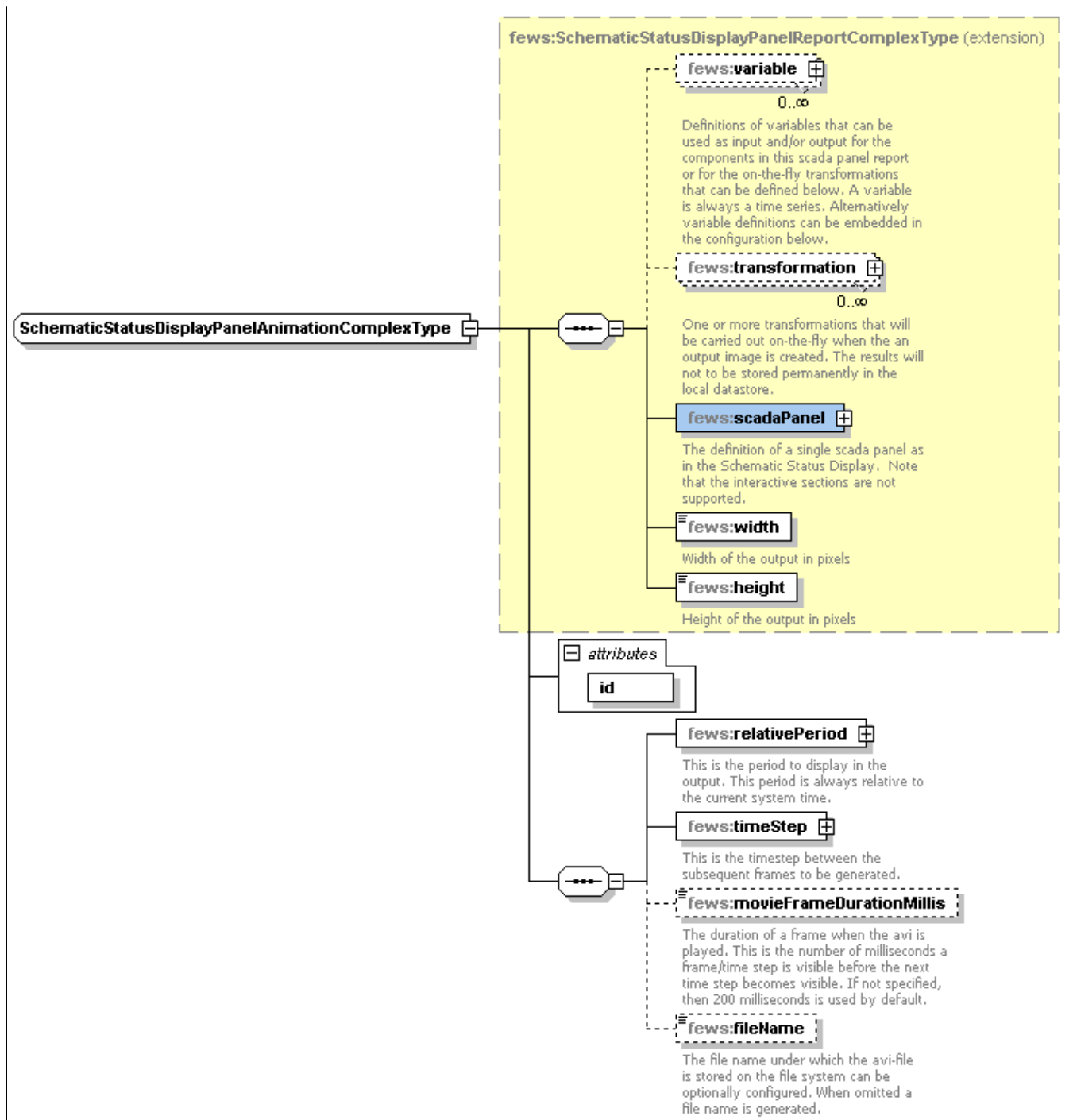
```
<schematicStatusDisplayPanelAvi id="ssdAvi">
        <scadaPanel id="TK" name="Twentekanalen 10 min">
                <svgFile>Twentekanalen.svg</svgFile>
                <nodeId>Twentekanalen Hydro</nodeId>
                ...
        </scadaPanel>
        <width>1024</width>
        <height>800</height>
        <relativePeriod unit="hour" start="-20" end="4"/>
        <timeStep unit="second" multiplier="300"/>
        <movieFrameDurationMillis>200</movieFrameDurationMillis>
        <fileName>SSD_test.avi</fileName>
</schematicStatusDisplayPanelAvi>
```

Animation in animated Gif format

With this function an animation of the configured period can be generated for a single Schematic Status Display panel. Use id $SSD_ANIMATED_GIF (myId)$ for including the animated gif file in the reports.

schematicStatusDisplayPanelAnimatedGif


```
<schematicStatusDisplayPanelAnimatedGif id="ssdGif">
        <scadaPanel id="TK" name="Twentekanalen 10 min">
                <svgFile>Twentekanalen.svg</svgFile>
                <nodeId>Twentekanalen Hydro</nodeId>
                ...
        </scadaPanel>
        <width>1024</width>
        <height>800</height>
```
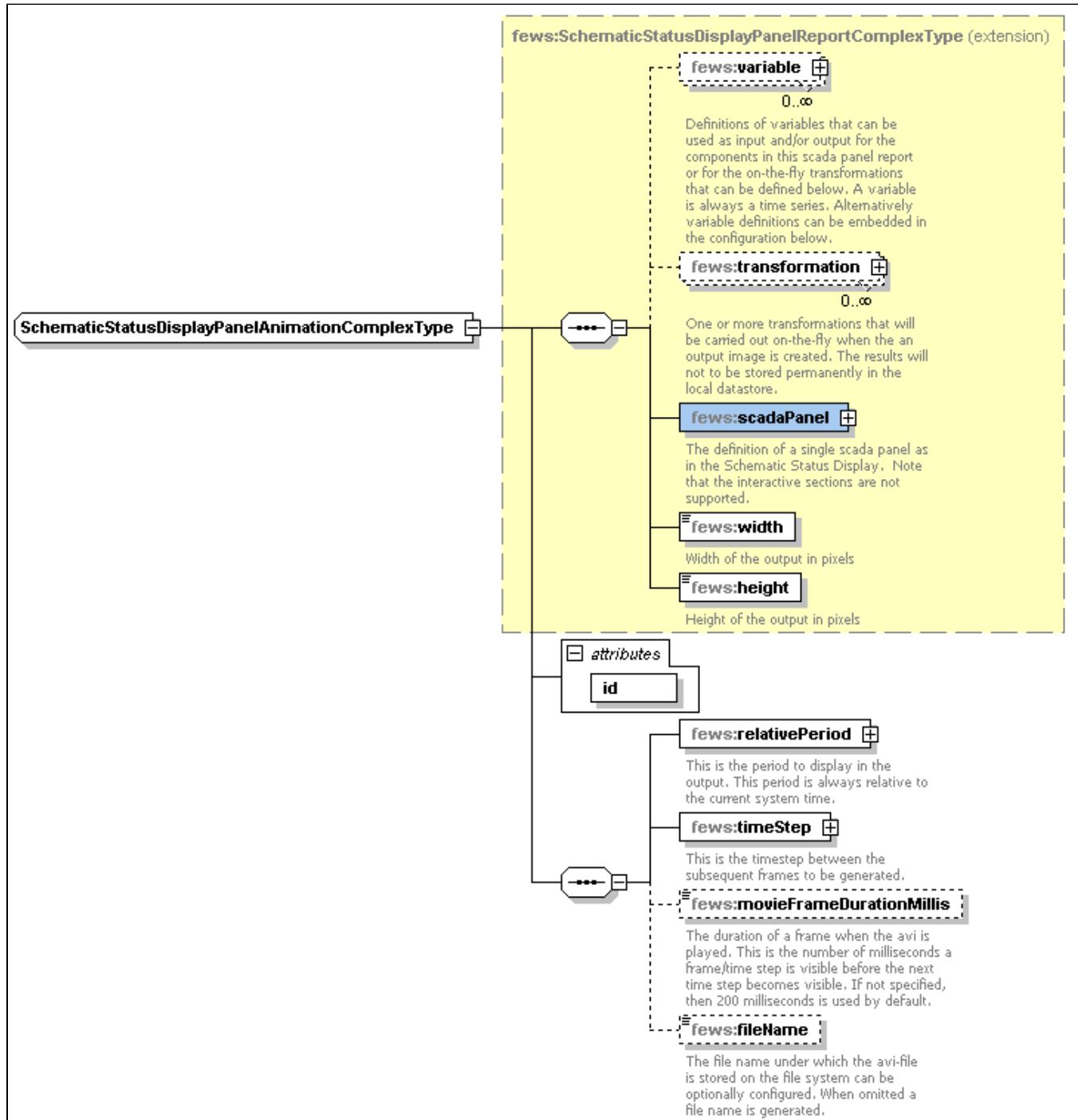
```
                        <relativePeriod unit="hour" start="-20" end="4"/>
                        <timeStep unit="second" multiplier="300"/>
                        <movieFrameDurationMillis>200</movieFrameDurationMillis>
                        <fileName>SSD_test.gif</fileName>
            </schematicStatusDisplayPanelAnimatedGif>
```
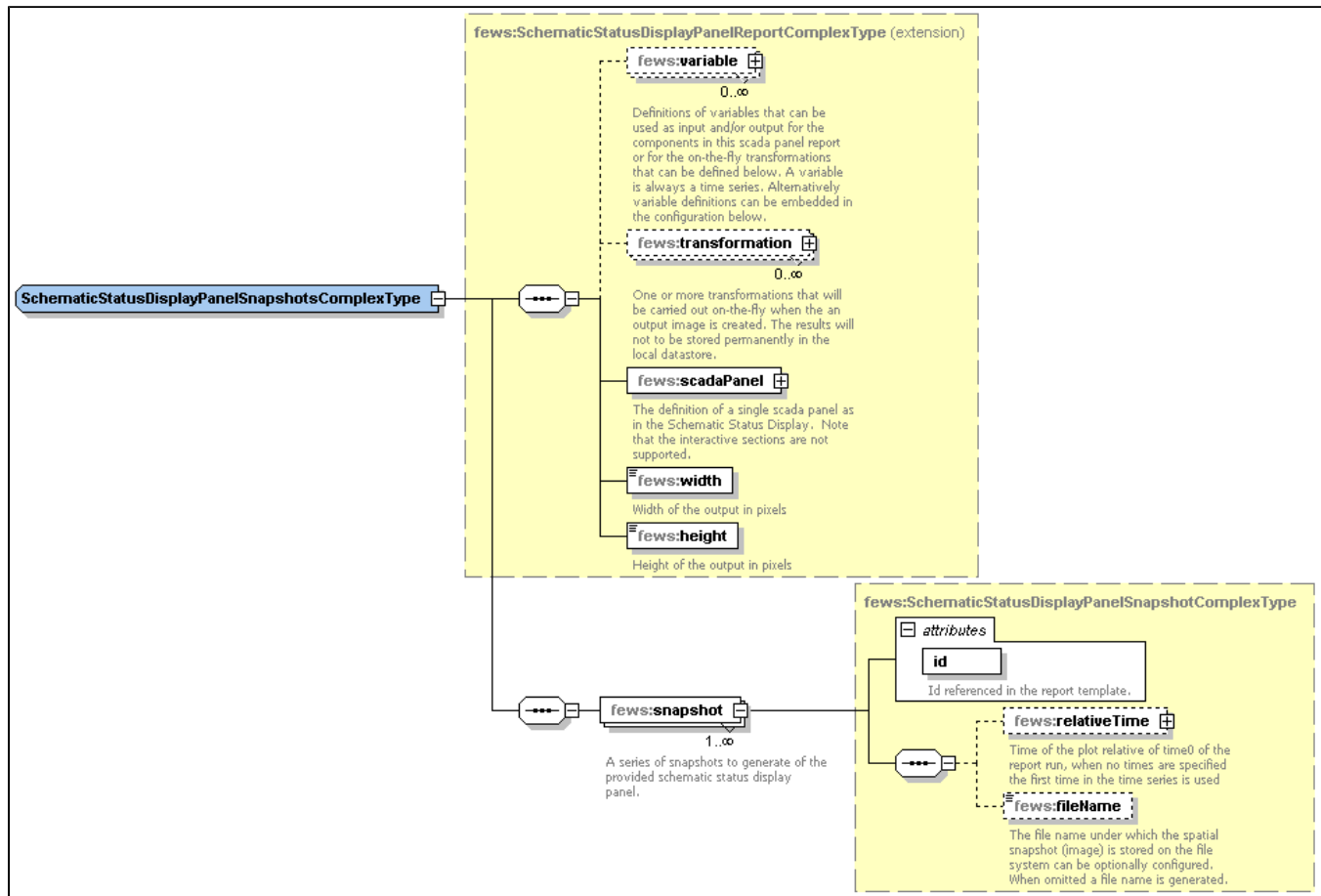
Snapshots in Png format

With this function one or more png images can be generated of a single Schematic Status Display panel. Use id $SSD_PNG(myId)$ for including the png files in the reports.

schematicStatusDisplayPanelSnapshotsPng

Sample demonstrating use of creating png snapshots that includes Schematic Status Display transformations:

```
        <schematicStatusDisplayPanelSnapshotsPng>
                <variable>
                        ...
                </variable>
                <transformation id="UserSimpleTransformation">
                        <user>
                                <simple>
                                        <expression>Waterstanden_10min * 100</expression>
                                        <outputVariable>
                                                <variableId>Waterstanden_10min_times_100</variableId>
                                        </outputVariable>
                                </simple>
                        </user>
                </transformation>

                <scadaPanel id="TK" name="Twentekanalen  10 min">
....
                </scadaPanel >
```

```
            <width>1024</width>
            <height>800</height>
            <snapshot id="ssd1">
                    <relativeTime unit="hour" value="-4"/>
                    <fileName>SSD_test1.png</fileName>
            </snapshot>
            <snapshot id="ssd2">
                    <relativeTime unit="hour" value="-3"/>
                    <fileName>SSD_test2.png</fileName>
            </snapshot>
    </schematicStatusDisplayPanelSnapshotsPng>
```
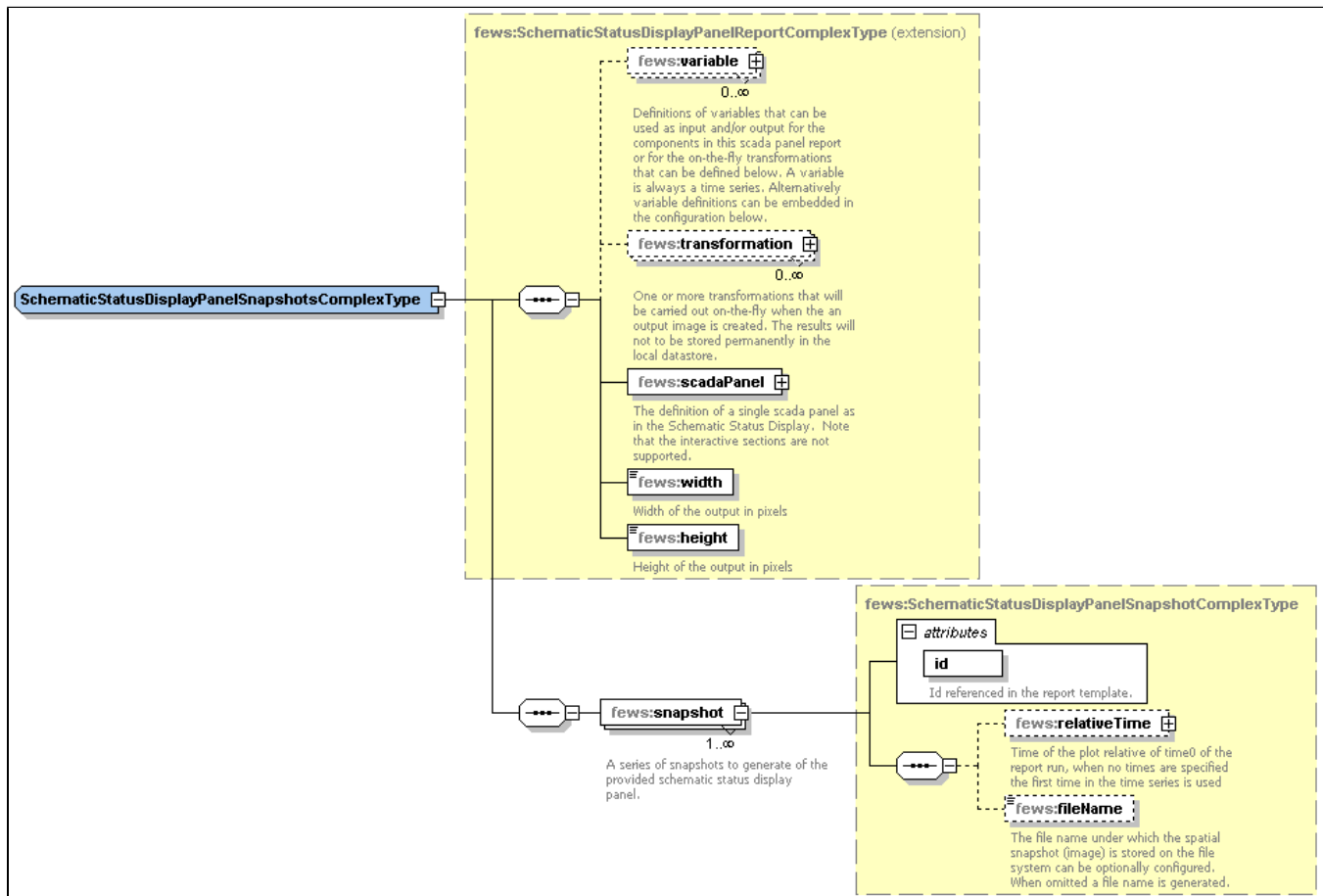
Snapshots in Svg format

With this function one or more svg images can be generated of a single Schematic Status Display panel. Use id $SSD_SVG(myId)$ for including the svg files in the reports.
The interactive parts of the Schematic Status Display such as Left Single Click Action and Left Double Click Action that can be configured when the Schematic Status Display is run in Delft-FEWS are not supported when exported as .svg report and it is recommended to remove these interactive sections where appropriate.

schematicStatusDisplayPanelSnapshotsSvg


```
        <schematicStatusDisplayPanelSnapshotsSvg>
                <scadaPanel id="TK" name="Twentekanalen  10 min">
....
                </scadaPanel >
                <width>1024</width>
                <height>800</height>
                <snapshot id="ssd1">
```

```
                        <relativeTime unit="hour" value="-4"/>
                        <fileName>SSD_test4.svg</fileName>
                </snapshot>
                <snapshot id="ssd2">
                        <relativeTime unit="hour" value="-3"/>
                        <fileName>SSD_test5.svg</fileName>
                </snapshot>
        </schematicStatusDisplayPanelSnapshotsSvg>
```

From the FEWS version **2018.02** it is possible to configure filename prefix and suffix for schematicStatusDisplayPanelSnapshotsPng and schematicStatusDisplayPanelSnapshotsSvg reports.

If fileNamePrefix is configured it will add the system time date to the beggining of file name in the configured format. If fileNameSuffix is configured it will add the system time date to end of the file name in the configured format.

```
<snapshot id="SchematicStatusSnapshot">
    <relativeTime unit="day" value="0"/>
    <fileName>scada.png</fileName>

    <fileNamePrefix>
            <timeZeroFormattingString>yyyy_</timeZeroFormattingString>
    </fileNamePrefix>

    <fileNameSuffix>
            <timeZeroFormattingString>MMDDHHmm</timeZeroFormattingString>
    </fileNameSuffix>
</snapshot>
```