

07 Filters

What	Required	Description	schema location
Filters.xml	yes	Definition of filters in the main map display	https://fewdocs.deltares.nl/schemas/version1.0/filters.xsd

Contents

- General
- filter
 - description
 - validationIconsViewPeriod and validationIconsVisible
 - ignoreDoubtfulAndUnreliableFlagSourceColumnId
 - visibilityControllingFlagSourceColumnId
 - onlyReliableFlagSourceColumnId
 - lastValueVisible
 - ignoreDeletionsOutsideViewPeriod
 - thresholdIconsBasedOnLastNonMissingValue
 - mapExtentId
 - viewPermission
 - tableViewPermission
 - editPermission
 - child
 - timeSeriesSet
 - timeSeriesSetId & constraints
 - Dynamic filter without without TimeSeriesSets
 - GroupBy
 - parameter attribute
 - Group by location and qualifier attributes
 - Nested GroupBy
 - Additional time series

General

Filters are used in DELFT-FEWS to define the locations that are displayed on the main map display, and that can be selected to display data. Filters are defined to arrange locations, with associated parameters in logical groups. Each filter is defined as a collection of time series sets.

Filters may be defined as a nested structure, allowing for the definition of a hierarchical set of filters.

When available on the file system, the name of the XML file is for example: Filters 1.00 default.xml

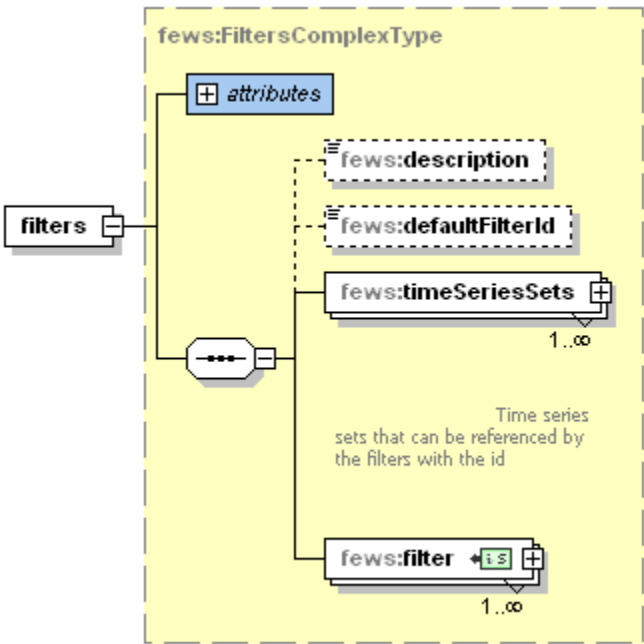


Figure 31 Elements of the filters configuration

The main elements of the filters configuration are shown in the image above and explained further below:

- **description:** Optional description of the filter configuration. Used for reference purposes only.
- **defaultFilterId:** Filter that is selected automatically on start up of FEWS. If not defined no filter will be selected.
- **timeSeriesSets:** Optional element that may appear an unlimited amount of times. It is possible to explicitly define every (child)filter. This may result in to many repeating timeSeriesSet definitions. Therefore it is also possible (since version 2009.02) to define groups of timeSeriesSets that can be used many times in the filter, additionally by using constraints on the location attributes (see [timeSeriesSetId & constraints](#)). Each <timeSeriesSets> element should contain an id attribute and one or more <timeSeriesSet> elements.
- **filter:** Required element with the definition of a filter. Multiple entries may exist where multiple filters are defined.

filter

Definition of a filter. Multiple entries may exist where multiple filters are defined. Each filter may contain either a set of one or more time series set, or a child filter. The child is a reference to another filter definition that again contains either a child filter or a list of time series sets. This structure is used to construct a hierarchal tree view of filters.

Attributes:

- **id:** Id of the filter. Used when this filter is referenced elsewhere in the configuration. Used in the tree view in the main display when no name attribute is given.
- **name:** optional name for the filter. When given it is used in the tree view in the main display.

The possible elements are each explained in their own section.

description

Optional description of the filter configuration. Used for reference purposes only.

validationIconsViewPeriod and validationIconsVisible

Both elements are optional, only one of them may be configured for a filter.

The validation icons are only visible when ALL the selected filters, including there children, have the validation period or validation icons visible configured.

- **validationIconsViewPeriod:** Since 2017.01, optional element. Calculate validation icons in the specified period. By using this you can for example ignore the validation problems for the most recent data.
- **validationIconsVisible:** Optional element allowing the user to make use of the additional validation icons available in the explorer (2009.01). When set to true, all validation icons in the view period of the time series sets will be shown.

ignoreDoubtfulAndUnreliableFlagSourceColumnId

Hide validation icons when this flag source column is filled.

Can be used to show all data that still needs to be validated, because then data that is validated can have this specific Flag Source Column filled.

Needs to be used in combination with either validationIconsViewPeriod or validationIconsVisible whcih are specified above

visibilityControllingFlagSourceColumnId

By defining a 'visibilityControllingFlagSourceColumnId' element, the existence of a flagSource Column for a particular timeseries (step) can be used as a condition for the data to be visible. When the configured flagSourceColumn is not existing for a particular timeseries step, the value will be perceived as 'Missing'. All values that did not pass this validation step are removed on read. They are set to missing for equidistant timeseries and removed for non-equidistant timeseries. The flagSourceColumnId needs to be defined in [flagSourceColumn.xml configuration file](#).

onlyReliableFlagSourceColumnId

By defining a 'onlyReliableFlagSourceColumnId' element, the existence of a flagSource Column for a particular timeseries (step) can be used as a condition for the data to be visible, but then only for reliable data. When the configured flagSourceColumn is not existing for a particular timeseries step, the value will be perceived as 'Missing'. All values that did not pass this validation step are removed on read. They are set to missing for equidistant timeseries and removed for non-equidistant timeseries. The flagSourceColumnId needs to be defined in [flagSourceColumn.xml configuration file](#). This works the same as the above element visibilityControllingFlagSourceColumnId but then only for reliable data, it can be used together with a visibilityControllingFlagSourceColumnId, then the data needs to have either the visibilityControllingFlagSourceColumnId (the quality flag does not matter) or it need to have the onlyReliableFlagSourceColumnId and the data needs to be reliable.

lastValueVisible

Optional element available since 2013.01.

In versions 2013.01 until 2017.01, when this filter is selected the last value for the selected parameter is visible in the location label on the map. If several filters are selected, the last value is only shown if all filters have this element set to true. When a parent filter is selected, the last values are only shown if the element is set to true for all child filters and the parent filter.

Since 2017.02 a button was added to the explorer which controls whether the labels show the last value. When a filter selection is made for which in previous versions the last values would be shown, the button is now turned on automatically instead. Note that this means the user can still turn of the last values by turning the button back off (clicking it). Also note that when a different selection is made (while the button is still on), the button will remain turned on even if <lastValueVisible> is not set to true for the new selection.

Note that in all versions, the last values are only shown for locations which do not have a custom label configured. Additionally, the values are only shown when a single parameter is selected in the data viewer. Finally, only values in the time span (relativeViewPeriod) configured for the timeSeriesSet will be found.

ignoreDeletionsOutsideViewPeriod

A light red cross is displayed when there is data available outside the view period. By default for performance reasons a red cross is still displayed when values once exist but are now deleted. When this option is set to false the complete available time series are read to take the deletions outside the view period into account.

thresholdIconsBasedOnLastNonMissingValue

Optional element, available since 2019.01, which shows the threshold icons based on the last non-missing value within the relative view period specified.

mapExtentId

Optional element, when this filter is selected the map will automatically zoom to the specified map extent.

viewPermission

Optional element, set a permission to control who can view this filter.

tableViewPermission

Since 2023.01. Optional element, set a permission to control who can view the actual data in this filter. Users without this permission cannot see the table data or export data. This permission also protects the data that is accessed via the display groups or time series lister

editPermission

Optional element, set a permission to control who can edit timeseries enumerated in this filter (only editable timeseries can be edited).

child

Reference to another filter. The child element refers to the ID of the other filter as a foreign key.

Attributes:

- **foreignKey**: Reference to ID of another filter, that is displayed as child filter

timeSeriesSet

Definition of a time series set belonging to a filter. Multiple time series sets may be defined.



The time span defined in the time series sets in the filter has an important function. It determines the time span checked for determining status icons (e.g. missing data) in the main map display.

Not all locations need to be included in a filter. Locations that are not defined, will never be visible to the user.

The readWrite parameter defined in the time series set included in the filter will determine if the time series may be edited by the user. If this parameter is set to read only then the time series will not support editing. This is the case for external time series only. Simulated time series are read only by convention. Notice that editable timeSeriesSets should have synchLevel 5 to get the edits saved into the central database.

```
<!--Example (time series sets not expanded) -->
  <filter id="Config Example">
    <child foreignKey="Nested Filter"/>
  </filter>
  <filter id="Nested Filter">
    <timeSeriesSet/>
  </filter>
  <filter id="Main Level Filter">
    <timeSeriesSet/>
  </filter>
```



Figure32 Example of filter configuration, as defined in the example XML configuration above.

timeSeriesSetId & constraints

Instead of an extensive definition of all possible time series that usually will be repeated for different types of locations and various districts, it is possible to define the timeSeriesSets once and to use it various times through all filters. At this point you refer to such a timeSeriesSets and optionally add some constraints on the location attributes. Notice that location attributes only can be defined in the locationSets for locations that are read from an ArcGIS shape DBF file. See the example below.

defaultFilterId main

timeSeriesSets (3)

	= id	timeSeriesSet
1	quantity	timeSeriesSet (33)
2	quality	timeSeriesSet (294)
3	groundwater	timeSeriesSet (4)

filter (1)

	= id	= name	mapExtentId	child								
1	main	Main system	full	child (3) <table><thead><tr><th></th><th>= foreignKey</th></tr></thead><tbody><tr><td>1</td><td>DischargePoints</td></tr><tr><td>2</td><td>PumpStations</td></tr><tr><td>3</td><td>Gauges</td></tr></tbody></table>		= foreignKey	1	DischargePoints	2	PumpStations	3	Gauges
	= foreignKey											
1	DischargePoints											
2	PumpStations											
3	Gauges											

filter (55)

	= id	= name	validationConsVisible	timeSeriesSetId	locationConstraints												
1	DischargePoints	Flows	false	quantity	locationConstraints <table><thead><tr><th colspan="2">attributeTextEquals (2)</th></tr><tr><th>= id</th><th>= equals</th></tr></thead><tbody><tr><td>1 regio</td><td>Main</td></tr><tr><td>2 type</td><td>Flow</td></tr></tbody></table>	attributeTextEquals (2)		= id	= equals	1 regio	Main	2 type	Flow				
attributeTextEquals (2)																	
= id	= equals																
1 regio	Main																
2 type	Flow																
2	PumpStations	Pumps	false	quantity	locationConstraints <table><thead><tr><th colspan="2">attributeTextEquals (2)</th></tr><tr><th>= id</th><th>= equals</th></tr></thead><tbody><tr><td>1 regio</td><td>Main</td></tr><tr><td>2 type</td><td>Pump</td></tr></tbody></table>	attributeTextEquals (2)		= id	= equals	1 regio	Main	2 type	Pump				
attributeTextEquals (2)																	
= id	= equals																
1 regio	Main																
2 type	Pump																
3	Gauges	Gauges	false	quantity	locationConstraints <table><thead><tr><th colspan="2">attributeTextEquals (2)</th></tr><tr><th>= id</th><th>= equals</th></tr></thead><tbody><tr><td>1 regio</td><td>Main</td></tr><tr><td>2 parameter</td><td>H.obs</td></tr></tbody></table> <table><thead><tr><th colspan="2">attributeExists</th></tr><tr><th>= id</th><th>PARENT_ID</th></tr></thead></table>	attributeTextEquals (2)		= id	= equals	1 regio	Main	2 parameter	H.obs	attributeExists		= id	PARENT_ID
attributeTextEquals (2)																	
= id	= equals																
1 regio	Main																
2 parameter	H.obs																
attributeExists																	
= id	PARENT_ID																

It is possible to define filters that may become empty: no locations comply with the constraints. These filters are not displayed. An advantage of this approach is that all possible filters can be defined once and automatically become visible when a location complies to the constraints.

Dynamic filter without without TimeSeriesSets

Without time series sets the filter only contains time series that are available in the database. For every filter FEWS walks through all the time series available in the database. When a time series is deleted it is removed from the filter on the next compact index files, this happens on a live system at least once a day.

The following filter includes all time series with module instance ImportFC.

```
<filter id="dynamic">
  <timeSeries>
    <moduleInstanceId>ImportFC</moduleInstanceId>
  </timeSeries>
  <relativeViewPeriod start="-10" end="0" unit="day"/>
</filter>
```

GroupBy

parameter attribute

Since 2016.01 filters can be automatically generated based on different parameter attribute values.

```
<filter id="groupByParameterAttribute">
  <timeSeries>
    <moduleInstanceId>ImportFC</moduleInstanceId>
  </timeSeries>
  <relativeViewPeriod start="-100000" end="0" unit="day"/>
  <parameterConstraints>
    <idStartsWith prefix=""/>
  </parameterConstraints>
  <groupBy>
    <parameterAttributeId>ATTRIBUTE_1</parameterAttributeId>
  </groupBy>
</filter>
```

This will work for multivalued attributes as well

```
<attributeFile>
  <csvFile>parameterAttributes.csv</csvFile>
  <parameterId>%ID%</parameterId>
  <attribute id="ATTRIBUTE_1">
    <text>%ATTRIBUTE_1%</text>
  </attribute>
  <attribute id="ATTRIBUTE_2">
    <text>%ATTRIBUTE_2%</text>
  </attribute>
</attributeFile>
</parametersCsvFile>
```

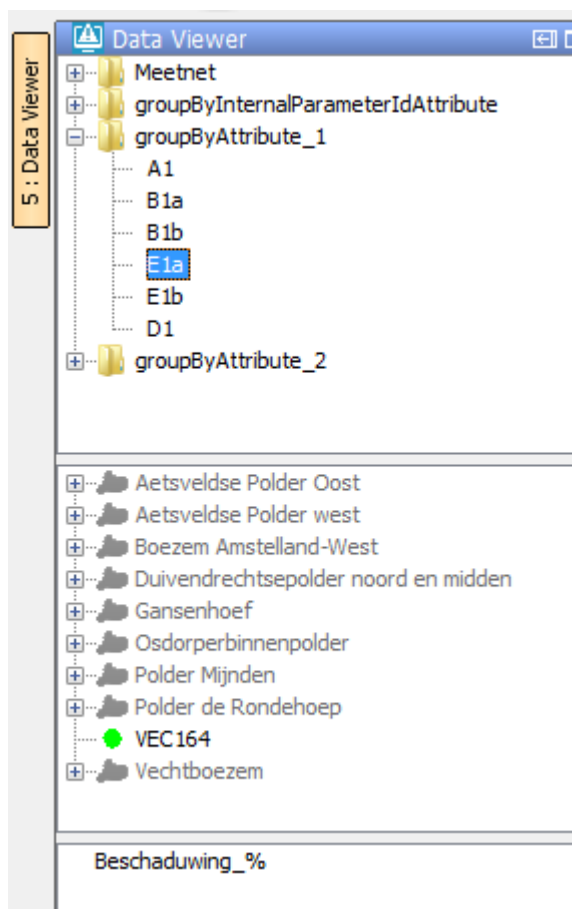
In parameterAttributes.csv below multivalued attributes are defined for some parameters like "BESDWG_%" which has values "E1a" & "E1b" for "ATTRIBUTE_1".

For parameter "BEMSRTJT_m" the attribute "ATTRIBUTE_1" has values "D1" & "E1b".

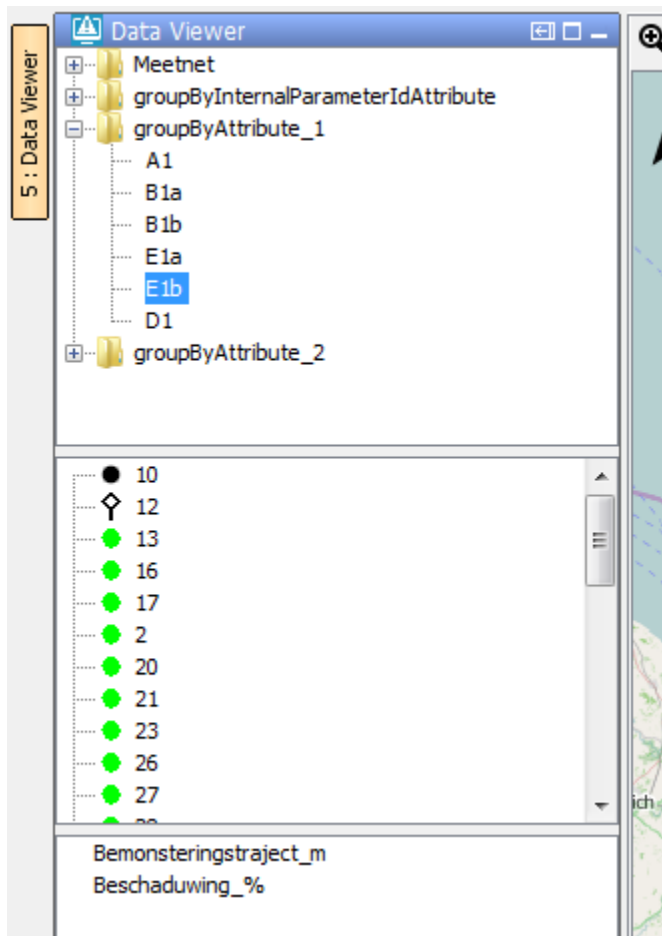
ID	ATTRIBUTE_1	ATTRIBUTE_2
BEMSDOVZDE	A1	A2
BEMSRBTE_m	B1b	B2
BEMSRPVK_m2		C2
Bemonsteringstraject_m	D1	
Beschaduwing_%	E1a	E2a
BREEDTE_m		
CHLFa_ug/l		
BEMSRBTE_m	B1a	
Beschaduwing_%	E1b	E2b
Bemonsteringstraject_m	E1b	

When "ATTRIBUTE_1" is used to generate filters based on its values the next filters are generated: A1, B1a, B1b, E1a, E1b, D1.

When selecting filter "E1a" only parameter "Beschaduwing" will appear because that is the only one with "E1a" as value for "ATTRIBUTE_1"



When selecting filter "E1b" both parameters "Beschaduwing_%" & "Bemonsteringstraject_m" will appear because they both have "E1b" in their multivalued "ATTRIBUTE_1":



Group by location and qualifier attributes

Since 2020.01 it is also possible to automatically generate filters based on location and qualifier attributes.

Nested GroupBy

GroupBy filters can be nested as well and multiple parameter, location and qualifier attributes can be used in any order and there is no limit on how many attributes to specify. However, creating too many of separate subfilters can result in a drastically degraded performance. Note that grouping by parameter attributes is available since 2016.01 while grouping by location attributes is available from version 2020.01.

The config example below will first create subfilters based on parameters which all have the same parameter attribute values.

The parameter filters will have subfilters based on location attributes and those subfilters will in turn have their own subfilters based on qualifier attribute values.

```
<filter id="GroupByNested" name="GroupByNested">
  <timeSeries>
    <moduleInstanceId>ImportFC</moduleInstanceId>
  </timeSeries>
  <relativeViewPeriod start="-31" end="0" unit="day"/>
  <groupBy>
    <parameterAttributeId>ParameterGroup</parameterAttributeId>
    <locationAttributeId>L_meetprogrammahistorie</locationAttributeId>
    <qualifierAttributeId>groep</qualifierAttributeId>
  </groupBy>
</filter>
```

Additional time series

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

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

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

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

When the attribute functions evaluate to the same value the additional time series are added as invisible by default. When additional time series are available the icon of the visibility dialog will change which shows that invisible time series are available and can be made visible. Each additional time series element can be given a name, this name will be used as prefix in the legend of the time series and as GroupNode label in the visibility dialog.

```
<filter id="GEBGEM_OPVL_PLGB_BEMALINGSGEBIED_DE_KEULEVAART" name="Peilbesluit bemalingsgebied De
Keulevaart">
  <mapExtentId>HDSR</mapExtentId>
  <timeSeriesSetsId>GEBGEM_OPVLWATER_PEILGEBIEDEN</timeSeriesSetsId>
  <locationConstraints>
    <attributeTextEquals id="PEILBESLUIT" equals="bemalingsgebied De Keulevaart"/>
  </locationConstraints>
  <additionalTimeSeries name="*O*">
    <locationFunctionEquals selected="@LOC_ID@" additional="@PEILGEBIED_ATT@"/>
    <timeSeriesSet>
      <moduleInstanceId>ImportOpvlWater</moduleInstanceId>
      <valueType>scalar</valueType>
      <parameterId>H.G.0</parameterId>
      <locationSetId>OPVLWATER_WATERSTANDEN</locationSetId>
      <timeSeriesType>external historical</timeSeriesType>
      <timeStep unit="nonequidistant"/>
      <relativeViewPeriod unit="day" start="-40" end="0"
        startOverrulable="false" endOverrulable="false"/>
      <readWriteMode>editing visible to all future task runs</readWriteMode>
      <synchLevel>1</synchLevel>
      <ensembleId>main</ensembleId>
      <ensembleMemberIndex>0</ensembleMemberIndex>
    </timeSeriesSet>
  </additionalTimeSeries>
  <additionalTimeSeries name="**O**">
    <locationFunctionEquals selected="@LOC_ID@" additional="@PEILGEBIED_ATT@"/>
    <timeSeriesSet>
      <moduleInstanceId>ImportOpvlWater</moduleInstanceId>
      <valueType>scalar</valueType>
      <parameterId>H.G.d</parameterId>
      <locationSetId>OPVLWATER_WATERSTANDEN</locationSetId>
      <timeSeriesType>external historical</timeSeriesType>
      <timeStep unit="day" timeZone="GMT+1"/>
      <relativeViewPeriod unit="day" start="-40" end="0"/>
      <readWriteMode>read only</readWriteMode>
    </timeSeriesSet>
  </additionalTimeSeries>
</filter>
```