

02 LocationSets

What	locationSets.xml
Required	no
Description	Definitions of groups of locations
schema location	http://fews.wldelft.nl/schemas/version1.0/locationSets.xsd

Contents

- [Overview](#)
- [Shape file, GeoJSON file and CSV file](#)
- [Database Table](#)
- [LocationSet XML file elements](#)
 - [locationSet](#)
 - [allowEmptyLocationSets](#)
 - [sortByName](#)
 - [sortingLocationAttributeld](#)
 - [chainageLocationAttributeld](#)
 - [chainageLabelLocationAttributeld](#)
 - [locationId](#)
 - [locationSetId](#)
 - [subLocationSetIdFunction](#)
 - [Label](#)

Overview

Location sets may be used to define logical groups of locations. Often an action may need to be taken on a whole set of locations (e.g. validation). By creating a LocationSet the action need only be defined once on a location set and not on individual locations. Any location may appear in more than one location set. Internally a location set is simply evaluated as a list of locations.

The location set configuration file is included in the RegionConfigFiles\ folder: LocationSets.xml

Shape file, GeoJSON file and CSV file

It is possible to define locationSets with locations that are automatically generated (so NOT defined in the locations.xml) from an ESRI Shape (dbf) file, geoJSON file or from a CSV table. See all detailed information at the [next page](#)

Database Table

It is also possible to define locationSets with locations that are read directly from a database. The contents of the database table are on the fly read and converted to a DBZ file. This DBZ file will be used by FEWS. This is for backup purpose in case the database is not available any more, like in stand-alone test environments. See all detailed information at the [next page](#)

LocationSet XML file elements

[locationSet](#)

Root element for the definition of a location set. Multiple entries may exist. The Id of the location set must be unique.

[allowEmptyLocationSets](#)

Optional element available since 2018.01. By default this element is set to true and FEWS will log a debug message when an empty locationSet is created. If set to false, FEWS will log a config error instead. When separate locationSets xml-files are used, each file can have its own <allowEmptyLocationSets> element, which will only apply for the location sets listed in that file.

[sortByName](#)

NOT used in dataviewer / filters. Sorting by name has become default, a sorting attribute can be defined in the location set <sortingLocationAttributeld>.

STILL used in exports

[sortingLocationAttributeld](#)

You can specify an attribute that should be used for sorting the locations in the user interface. This is useful for list locations in for example a logical river order (upstream to downstream). The attribute should then contain a numbering. Note that the attributes are ordered as strings, so use 00001 instead of 1. Each location can only have 1 attribute which defines the sorting, this will automatically be the first defined in locationSets.xml. Sorting of locations with different sorting attributes will result in grouping locations with the same sorting attribute together.

Some best practices when using the <sortingLocationAttributeId>:

- Give the main locationSet based on the csvFile a sortingLocationAttributeId that makes sense (e.g. Name or logical river order). This will be the first locationSet found by FEWS, and therefore will dictate the sorting order used in the [Filter](#) (aka DataViewer, configured in Filters.xml). This is because by default locations sets use the sorting of the parent location set.
- Give **all** locationSets derived from the main locationSet a sortingOrder to overrule the default by name sorting order.
- Create separate locationSets for displaysGroups and Exports when the order in these two is different.
- Don't define numeric attributes as text attributes in your configuration. Don't use leading zeros for numeric attributes in your csv. Leading zeros and/or text attributes will significantly increase the memory usage and start-up time of FEWS.
- (General recommendation) After making these kind of changes to config, if you experience unexpected behaviour in Delft-FEWS after a refresh, restart Delft-FEWS to see if this behaviour is persistent.
- In the filters locations with the same sorting mechanism (same attribute or by name) are grouped together (the order of grouping can depend on Java version)

Option is available since 2014.01

chainageLocationAttributeId

By configuring <chainageLocationAttributeId> all locations that have a value for this attribute will be contained in this location set. This attribute will automatically be used as sortingLocationAttributeId so the locations are in the order of ascending chainage value. The attribute must be of type numeric.

csv attribute file <pre>LOCID;CHAINAGE_C;CHAINAGE_D;CHAINAGE_E loc_a;0;; loc_b;150;0; loc_c;450;300; loc_d;700;550; loc_e;;;0 loc_f;;;50 loc_g;;;150 loc_h;;;300</pre>
LocationSets - reference to csv attribute file <pre><attributeFile> <csvFile>CHAINAGE_CDE.csv</csvFile> <id>%ID%</id> <timeZoneOffset>+00:00</timeZoneOffset> <attribute id="CHAINAGE_C"> <text>%CHAINAGE_C%</text> </attribute> <attribute id="CHAINAGE_D"> <text>%CHAINAGE_D%</text> </attribute> <attribute id="CHAINAGE_E"> <text>%CHAINAGE_E%</text> </attribute> </attributeFile></pre>
LocationSets - defenition of location sets <pre><locationSet id="CHAINAGE_D"> <chainageLocationAttributeId>CHAINAGE_D</chainageLocationAttributeId> </locationSet></pre>

use of locationSet to display data

```
<timeSeriesSet>
  <moduleInstanceId>ExportRunMultipleTimeSeries</moduleInstanceId>
  <valueType>scalar</valueType>
  <parameterId>H.m</parameterId>
  <chainageLocationSetId>CHAINAGE_D</chainageLocationSetId>
  <timeSeriesType>external historical</timeSeriesType>
  <timeStep unit="day"/>
  <relativeViewPeriod unit="day" start="-7" end="0"/>
  <readWriteMode>read only</readWriteMode>
</timeSeriesSet>
```

Since 2020.02 it is possible to use time dependent chainage location attributes see [Time Dependent Chainage Location Sets](#).

For configuring location attributes see all detailed information at the [next page](#)

Referring to the chainage location set can only be done in combination with <readWriteMode>read only</readWriteMode> by configuring <chainageLocationSetId> in <timeSeriesSet>, which is shown at the display groups page:

Display groups chainage location set

chainageLabelLocationAttributeId

This is an optional configuration field. It can be only configured together with chainageLocationAttributeId. If configured, this location attribute will be used as the label of vertical station lines in longitudinal plot. If a location does not have this attribute, there will be no vertical station line drawn. This should be a text attribute.

```
      <locationSet id="CHAINAGE_C">
        <chainageLocationAttributeId>Chainage</chainageLocationAttributeId>
        <chainageLabelLocationAttributeId>AddLabel</chainageLabelLocationAttributeId>
      .....
      .....
    </locationSet>
```

Location example:

```
<locations xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.wldelft.nl/fews" xsi:
schemaLocation="http://www.wldelft.nl/fews http://fews.wldelft.nl/schemas/version1.0/locations.xsd"
  xmlns:textAttribute="http://www.wldelft.nl/fews/textAttribute"
  xmlns:numberAttribute="http://www.wldelft.nl/fews/numberAttribute"
  xmlns:booleanAttribute="http://www.wldelft.nl/fews/booleanAttribute"
  version="1.1">
  <geoDatum>WGS 1984</geoDatum>

  <location id="LocA" name="LocA">
    <x>4.5</x>
    <y>53.5</y>
    <z>0</z>
    <textAttribute:AddLabel>This is region Europe</textAttribute:AddLabel>
  </location>
  <location id="LocB" name="LocB">
    <x>4.1</x>
    <y>53.1</y>
    <z>0</z>
  </location>
</locations>
```

In this example, both location A and B will appear on the longitudinal plot, but only location A will have a station line drawn (with label text "This is region Europe"). It is also possible to use this option with CSV or esriShape files.

description

Optional description of the location set. Used for reference purposes only.

locationId

Location ID configured to be a member of the locationSet. Multiple entries may exist.

locationSetId

LocationSet ID configured to be a member of the locationSet. Multiple entries may exist. This is useful to group locationSets together.

subLocationSetIdFunction

This element can be used to generate multiple locationsets from location attributes; these are normally invluded for locations in dbf or CSV files. In the example below a locationSet is created **Warrington_Fluvial_Catchments_PDM** from locationSet **UK_Fluvial_Catchments_PDM** where the location attribute *Centre* contains *Warrington*. By using the subLocationSetIdFunction, multiple locationSets are generated as well for all river_segments that can be found in the locationattribute @River_Segment@.

```
<locationSet id="Warrington_Fluvial_Catchments_PDM">
  <subLocationSetIdFunction>@River_Segment@_Warrington_Model_PDM</subLocationSetIdFunction>
  <locationSetId>UK_Fluvial_Catchments_PDM</locationSetId>
  <constraints>
    <attributeTextContains id="Centre" contains="Warrington"/>
  </constraints>
</locationSet>
```

With the XML example above, 2 river segment locationSets will be generated: Darwen_Warrington_Model_PDM (with 2 locations) and Dee_ISIS_Warrington_Model_PDM (with 3 locations).

LocationId	Name	River_Segment
713119	Harrison Street PDM	Darwen
713120	Ewood Blackburn PDM	Darwen
67008	Pont Y Capel PDM	Dee_ISIS
67015	Manley Hall PDM	Dee_ISIS
67021	Lea Hall PDM	Dee_ISIS

Note that sublocationsets can not be added to other locationsets. So it is advised to create an overarching locationset (like UK_Fluvial_Catchments_PDM) from which sublocationsets are created by the function.

Label

It is possible to define a label that is presented in the map. For more information, see [Explorer](#).