

How to Import data

Contents

- [Contents](#)
- [Summary](#)
- [Setting up the import module](#)
 - [Step 1: Configure the import module \(create the Import 1.00 default.xml configuration File\)](#)
 - [Step 2: Register the new module - Add to the ModuleInstanceDescriptors 1.00 default.xml file](#)
 - [Step 3: Add the module to a workflow \(ImportExternal 1.00 Default.xml\)](#)
 - [Step 4: Make the locationSet](#)
- [Set up Id Mapping](#)
 - [Step 5: Make the idmap file IdImport 1.00.xml](#)
- [Setting up a \(temporary\) display to view the imported data](#)
- [Add to the appropriate filters\(s\)](#)
- [Test the import](#)

Summary

This document contains a step by step guide how to import data in [PI XML](#) format into Delft-Fews.



Note: This document does *NOT* describe all the possible options of the Import module; this information can be found on this wiki page [03 Import Module](#).

Importing data in DELFT-FEWS is different from running a wizard in e.g. a much used office software package. To be able to import data a number of xml configuration files needs to be adjusted or created. In short, this entails the following steps:

1. The file-type you want to import should be readable by [one of the imports available in Delft-Fews](#)
2. You must configure an instance of this import module
3. You must add the moduleInstance to a workflow (usually the ImportExternal Workflow)
4. In case the locationId's and parameterId's in your Delft-Fews configuration are not identical to those in the files you need to import a IdMap file should be set up and referred to in the import module
5. You will probably need to adjust one or more locationsets, the filters.xml file and the DisplayGroups.xml file if you need to display or process the newly imported data.



Please note that Delft-Fews *DELETES* all files after importing them. As such, you should *ALWAYS* copy files from another directory to the location from which Delft-Fews imports files. If you do not do this you might loose you files.

Setting up the import module

Setting up the Import module to import data in PI XML format involves the following steps:

1. make the Configuration file for the import module and store in the ModuleConfigFiles directory. In this example it should be named **Import 1.00 default.xml**
2. register this new module in the **ModuleInstanceDescriptors 1.00 default.xml** file in the RegionConfigFiles directory
3. Add the new module at a workflow in order to run it.

Step 1: Configure the import module (create the [Import 1.00 default.xml](#) configuration File)

The first step is to configure an instance of the import module for the type of data you need to import. In this example we will import data in [PI XML](#) format for 4 locations. The [file we want to import](#) is shown below:

```

<?xml version="1.0" encoding="UTF-8"?>
<TimeSeries xmlns="http://www.wldelft.nl/fews/PI"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.wldelft.nl/fews/PI
http://fews.wldelft.nl/schemas/version1.0/pi-schemas/pi_timeseries.xsd"
version="1.2">
  <series>
    <header>
      <type>instantaneous</type>
      <locationId>LocA</locationId>
      <parameterId>WaterLevel</parameterId>
      <timeStep unit="second" multiplier="3600"/>
      <startDate date="2006-08-23" time="17:00:00"/>
      <endDate date="2006-08-24" time="00:00:00"/>
      <missVal>-8888.0</missVal>
      <longName>LocA</longName>
      <units>m</units>
    </header>
    <event date="2006-08-23" time="17:00:00" value="8.66"/>
    <event date="2006-08-23" time="18:00:00" value="9.66"/>
    <event date="2006-08-23" time="19:00:00" value="8.66"/>
    <event date="2006-08-23" time="20:00:00" value="-8888.0"/>
    <event date="2006-08-23" time="21:00:00" value="3"/>
    <event date="2006-08-23" time="22:00:00" value="3"/>
    <event date="2006-08-23" time="23:00:00" value="3"/>
    <event date="2006-08-24" time="00:00:00" value="-8888.0"/>
  </series>
  <series>
    <header>
      <type>instantaneous</type>
      <locationId>LocB</locationId>
      <parameterId>WaterLevel</parameterId>
      <timeStep unit="second" multiplier="3600"/>
      <startDate date="2006-08-23" time="13:00:00"/>
      <endDate date="2006-08-24" time="00:00:00"/>
      <missVal>-999.0</missVal>
      <longName>LocB</longName>
      <units>m</units>
    </header>
    <event date="2006-08-23" time="13:00:00" value="0.63"/>
    <event date="2006-08-23" time="14:00:00" value="0.61"/>
    <event date="2006-08-23" time="15:00:00" value="0.59"/>
    <event date="2006-08-23" time="16:00:00" value="0.59"/>
    <event date="2006-08-23" time="17:00:00" value="0.59"/>
    <event date="2006-08-23" time="18:00:00" value="0.59"/>
    <event date="2006-08-23" time="19:00:00" value="0.60"/>
    <event date="2006-08-23" time="20:00:00" value="0.57"/>
    <event date="2006-08-23" time="21:00:00" value="0.59"/>
    <event date="2006-08-23" time="22:00:00" value="0.59"/>
    <event date="2006-08-23" time="23:00:00" value="-999.0"/>
    <event date="2006-08-24" time="00:00:00" value="-999.0"/>
  </series>
  <series>
    <header>
      <type>instantaneous</type>
      <locationId>LocC</locationId>
      <parameterId>WaterLevel</parameterId>
      <timeStep unit="second" multiplier="3600"/>
      <startDate date="2006-08-23" time="13:00:00"/>
      <endDate date="2006-08-24" time="00:00:00"/>
      <missVal>-999.0</missVal>
      <longName>LocC</longName>
      <units>m</units>
    </header>
    <event date="2006-08-23" time="13:00:00" value="-1.42"/>
    <event date="2006-08-23" time="14:00:00" value="-1.42"/>
    <event date="2006-08-23" time="15:00:00" value="-1.44"/>
  </series>
</TimeSeries>

```

```

        <event date="2006-08-23" time="16:00:00" value="-1.46"/>
        <event date="2006-08-23" time="17:00:00" value="-1.46"/>
        <event date="2006-08-23" time="18:00:00" value="-1.46"/>
        <event date="2006-08-23" time="19:00:00" value="-1.48"/>
        <event date="2006-08-23" time="20:00:00" value="-1.48"/>
        <event date="2006-08-23" time="21:00:00" value="-1.48"/>
        <event date="2006-08-23" time="22:00:00" value="-1.49"/>
        <event date="2006-08-23" time="23:00:00" value="-999.0"/>
        <event date="2006-08-24" time="00:00:00" value="-999.0"/>
    </series>
    <series>
        <header>
            <type>instantaneous</type>
            <locationId>LocD</locationId>
            <parameterId>WaterLevel</parameterId>
            <timeStep unit="second" multiplier="3600"/>
            <startDate date="2006-08-23" time="13:00:00"/>
            <endDate date="2006-08-24" time="00:00:00"/>
            <missVal>-999.0</missVal>
            <longName>LocD</longName>
            <units>m</units>
        </header>
        <event date="2006-08-23" time="13:00:00" value="-1.42"/>
        <event date="2006-08-23" time="14:00:00" value="-1.42"/>
        <event date="2006-08-23" time="15:00:00" value="-1.44"/>
        <event date="2006-08-23" time="16:00:00" value="-1.46"/>
        <event date="2006-08-23" time="17:00:00" value="-1.46"/>
        <event date="2006-08-23" time="18:00:00" value="-1.46"/>
        <event date="2006-08-23" time="19:00:00" value="-1.48"/>
        <event date="2006-08-23" time="20:00:00" value="-1.48"/>
        <event date="2006-08-23" time="21:00:00" value="-1.48"/>
        <event date="2006-08-23" time="22:00:00" value="-1.49"/>
        <event date="2006-08-23" time="23:00:00" value="-999.0"/>
        <event date="2006-08-24" time="00:00:00" value="-999.0"/>
    </series>
</TimeSeries>

```

When configuring the import module we will need to get the following information from the file(s) to be imported:

1. the locationId's in the XML file
2. the parameterId's in the XML file
3. the timestep in the XML file

This information is needed to configure the timeseriesSet in the import module. A quick look at the file shows that there are four locations: LocA, LocB, LocC and LocD. Only one parameter is present (WaterLevel) and the data comes in hourly timesteps.

Another type of information that we will need is the location at which Delft-Fews can find the files to be imported. For this example we assume the files will be located in c:/Import/PI .

With this we can make the [file below](#):



Please see the Delft-Fews configuration guide for detailed information on how to configure the Import modules. In addition, the .xsd schema may be consulted.

```

<?xml version="1.0" encoding="UTF-8"?>
<timeSeriesImportRun xmlns="http://www.wldelft.nl/fews"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.wldelft.nl/fews
http://fews.wldelft.nl/schemas/version1.0/timeSeriesImportRun.xsd">
  <import>
    <general>
      <importType>PI</importType> <!-- Indicated we will read PI Files -->
      <folder>c:/Import/PI</folder> <!-- read the files from here -->
      <!-- Files WILL BE DELETED after import -->
      <failedFolder>c:/Backup/PI</failedFolder>
      <!-- If specified the imported files will be copied to this location -->
      <idMapId>IdImport</idMapId> <!-- The IdMap for this import -->
      <importTimeZone>
        <timeZoneOffset>+00:00</timeZoneOffset>
      </importTimeZone>
    </general>
    <timeSeriesSet>
      <moduleInstanceId>Import</moduleInstanceId> <!-- The name of this moduleinstance -->
      <valueType>scalar</valueType>
      <parameterId>H.obs</parameterId> <!-- The parameter as it will be stored in Delft-Fews,
      different from the parameter in the XML. Therefore,
      IdMapping has to be set up-->
      <locationSetId>LevelGauges</locationSetId>
      <timeSeriesType>external historical</timeSeriesType>
      <timeStep unit="hour" multiplier="1"/>
      <readWriteMode>add originals</readWriteMode>
      <synchLevel>1</synchLevel>
    </timeSeriesSet>
  </import>
</timeSeriesImportRun>

```

In the general section of the configuration the location at which the file(s) are to be found is given in addition to an optional backup location. Imported data will be copied to the backup location if it is defined and accessible. All imported files in the **folder** location will be **deleted**. The contents of the **idMapId** element refers to the IdMapping table that converts the external locations/parameters to those used in your Delft-Fews application. If the mapping is one-to-one the idmap is not needed.

The **timeSeriesSet** element defines how the data will be stored in Delft-Fews. If you later want to view, use or export the data you will need the same timeseriesset to retrieve or view the data. In the timeseriesset we have defined the following:

- moduleInstanceId - Name of this configuration instance, reflects the XML file name
- valueType - scalar, this is scalar data
- parameterId - H.obs - This is the parameterId in Delft-Fews. The parameter Id in the XML file is WaterLevel. The Id mapping (shown below) will map Waterlevel to H.obs
- locationSetId - In order to simplify configuration we have defined a list of locations (a locationset) in the file [LocationSets.xml](#) (in the RegionConfigFiles directory). This set holds the following four locations: one, two, three, four. As these are different from the locations in the XML file to be imported id mapping will be set up.
- timeStep - the timestep defined here should map the timestep in the XML files to be imported.

Step 2: Register the new module - Add to the [ModuleInstanceDescriptors 1.00 default.xml](#) file

All moduleInstances must be registered in Delft-Fews. This is done by adding an entry in the moduleInstanceDescriptors file in the RegionConfigFiles section of the configuration. The [moduleInstancedescriptors](#) file shown below includes this appropriate entry:



In this example only one moduleInstance is present. Usually you will need to add your module to an existing list.

```
<?xml version="1.0" encoding="UTF-8"?>
<moduleInstanceDescriptors xmlns="http://www.wldelft.nl/fews"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.wldelft.nl/fews
http://fews.wldelft.nl/schemas/version1.0/moduleInstanceDescriptors.xsd"
version="1.0">
  <!--Import Module Instances-->
  <moduleInstanceDescriptor id="Import">
    <description>Import Our new data</description>
    <moduleId>TimeSeriesImportRun</moduleId>
  </moduleInstanceDescriptor>
</moduleInstanceDescriptors>
```

Step 3: Add the module to a workflow ([ImportExternal 1.00 Default.xml](#))

In order to use the module (to be able to run it from the manual forecast dialog) you will need to list the module as an activity in a workflow. Assuming this workflow is already registered (see the Delft-Fews configuration manual on how to register workflow) the only thing to do is to add in to the workflow. In the following example the new module is the only activity in the workflow:

```
<?xml version="1.0" encoding="UTF-8"?>
<workflow xmlns="http://www.wldelft.nl/fews"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.wldelft.nl/fews[http://fews.wldelft.nl/schemas/version1.0/workflow.xsd]"
version="1.1">
  <activity>
    <runIndependent>true</runIndependent>
    <moduleId>Import</moduleId>
  </activity>
</workflow>
```

Step 4: Make the locationSet

In order to refer to all four locations in one go (Instead of having to create four different timeseriesset) a locationset is defined in the file [LocationSets 1.00 Default.xml](#) in the RegionConfigFiles directory:

```
<?xml version="1.0" encoding="UTF-8"?>
<locationSets version="1.1" xmlns="http://www.wldelft.nl/fews" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemaLocation="http://www.wldelft.nl/fews[http://fews.wldelft.nl/schemas/version1.0/locationSets.
xsd]">
  <locationSet id="LevelGauges" name="LevelGauges">
    <locationId>one</locationId>
    <locationId>two</locationId>
    <locationId>three</locationId>
    <locationId>four</locationId>
  </locationSet>
</locationSets>
```

Set up Id Mapping

Step 5: Make the idmap file [IdImport 1.00.xml](#)

The format of the IdMap files is described in the DELFT-FEWS configuration guid and also in the .xsd scheme. All idMaps are stored in the IdMapFiles directory. The Idmap needed for this example ([IdImport 1.00 Default.xml](#)) should match the parameter WaterLevel to H.obs and should also map the four locations. The files is shown below:

```
<?xml version="1.0" encoding="UTF-8"?>
<idMap xmlns="http://www.wldelft.nl/fews"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.wldelft.nl/fews
http://fews.wldelft.nl/schemas/version1.0/idMap.xsd" version="1.1">
  <parameter internal="H.obs" external="WaterLevel"/>
  <location internal="one" external="LocA"/>
  <location internal="two" external="LocB"/>
  <location internal="three" external="LocC"/>
  <location internal="four" external="LocD"/>
</idMap>
```

Setting up a (temporary) display to view the imported data

Delft-Fews only shows data that is configured to be visible to the user. All other data is invisible to the user. The file [DisplayGroups.xml](#) (located in the SystemConfigFiles directory) defines pre-defined graphs. In the example below five graphs have been defined to view the newly imported data:

```
<?xml version="1.0" encoding="UTF-8"?>
<displayGroups version="1.0"
xmlns="http://www.wldelft.nl/fews"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.wldelft.nl/fews
http://fews.wldelft.nl/schemas/version1.0/displayGroups.xsd">
  <displayGroup name="Test">
    <display name="Gauge One">
      <relativeViewPeriod unit="hour" start="-12" end="36"/>
      <subplot>
        <timeSeriesSet>
          <moduleInstanceId>Import</moduleInstanceId>
          <valueType>scalar</valueType>
          <parameterId>H.obs</parameterId>
          <locationId>one</locationId>
          <timeSeriesType>external historical</timeSeriesType>
          <timeStep unit="hour" multiplier="1"/>
          <relativeViewPeriod unit="hour" end="48" start="-48"/>
          <readWriteMode>add originals</readWriteMode>
        </timeSeriesSet>
      </subplot>
    </display>
    <display name="Gauge Two">
      <relativeViewPeriod unit="hour" start="-12" end="36"/>
      <subplot>
        <timeSeriesSet>
          <moduleInstanceId>Import</moduleInstanceId>
          <valueType>scalar</valueType>
          <parameterId>H.obs</parameterId>
```

```

        <locationId>two</locationId>
        <timeSeriesType>external historical</timeSeriesType>
        <timeStep unit="hour" multiplier="1"/>
        <relativeViewPeriod unit="hour" end="48" start="-48"/>
        <readWriteMode>add originals</readWriteMode>
    </timeSeriesSet>
</subplot>
</display>
<display name="Gauge Three">
    <relativeViewPeriod unit="hour" start="-12" end="36"/>
    <subplot>
        <timeSeriesSet>
            <moduleInstanceId>Import</moduleInstanceId>
            <valueType>scalar</valueType>
            <parameterId>H.obs</parameterId>
            <locationId>three</locationId>
            <timeSeriesType>external historical</timeSeriesType>
            <timeStep unit="hour" multiplier="1"/>
            <relativeViewPeriod unit="hour" end="48" start="-48"/>
            <readWriteMode>add originals</readWriteMode>
        </timeSeriesSet>
    </subplot>
</display>
<display name="Gauge Four">
    <relativeViewPeriod unit="hour" start="-12" end="36"/>
    <subplot>
        <timeSeriesSet>
            <moduleInstanceId>Import</moduleInstanceId>
            <valueType>scalar</valueType>
            <parameterId>H.obs</parameterId>
            <locationId>four</locationId>
            <timeSeriesType>external historical</timeSeriesType>
            <timeStep unit="hour" multiplier="1"/>
            <relativeViewPeriod unit="hour" end="48" start="-48"/>
            <readWriteMode>add originals</readWriteMode>
        </timeSeriesSet>
    </subplot>
</display>
<display name="All together">
    <relativeViewPeriod unit="hour" start="-12" end="36"/>
    <subplot>
        <timeSeriesSet>
            <moduleInstanceId>Import</moduleInstanceId>
            <valueType>scalar</valueType>
            <parameterId>H.obs</parameterId>
            <locationSetId>LevelGauges</locationSetId>
            <timeSeriesType>external historical</timeSeriesType>
            <timeStep unit="hour" multiplier="1"/>
            <relativeViewPeriod unit="hour" end="48" start="-48"/>
            <readWriteMode>add originals</readWriteMode>
        </timeSeriesSet>
    </subplot>
</display>
</displayGroup>
</displayGroups>

```

Add to the appropriate filters(s)

To be able to use the main map display and the filters to access the data you should set up an entry in the [Filters.xml](#) file in the RegionConfigFiles directory. See the example below:

```
<?xml version="1.0" encoding="UTF-8"?>
<filters version="1.1"
xmlns="http://www.wldelft.nl/fews"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.wldelft.nl/fews
http://fews.wldelft.nl/schemas/version1.0/filters.xsd">
  <filter name="Gauges" id="Gauges">
    <timeSeriesSet>
      <moduleInstanceId>Import</moduleInstanceId>
      <valueType>scalar</valueType>
      <parameterId>H.obs</parameterId>
      <locationSetId>LevelGauges</locationSetId>
      <timeSeriesType>external historical</timeSeriesType>
      <timeStep unit="hour" multiplier="1"/>
      <relativeViewPeriod unit="hour" end="48" start="-48"/>
      <readWriteMode>add originals</readWriteMode>
    </timeSeriesSet>
  </filter>
</filters>
```

Test the import

To test the import run the ImportExternal Workflow. Check to log panel for error messages. If you want to view the data please make sure to first set the system time to a time/date you have data for. In this example August 23 2006.