

# 02 Module Parameters

## Module Parameters

Module Parameters can also be managed by DELFT-FEWS similar to management of Module datasets. The difference is that where the module datasets are handled as ZIP files, with no actual interaction between the dataset and DELFT-FEWS, module parameters can be defined in a native DELFT-FEWS format and exchanged with the external module in the published interface format through the General Adapter module. A prerequisite for this exchange being meaningful is that the module adapter supports this format of the published interface, and can transform this into the native module format.

As in the module datasets, module parameters are defined in a configuration where the name is the same as the moduleInstanceId of the General Adapter module it is to be used in (though a different name may also be called by the General Adapter- see [05 General Adapter Module#05GeneralAdapterModule-exportParameterActivity](#)).

When available on the file system, the name of the XML file for configuring Module Parameters for example for the Eden\_Historical module may be:

Eden\_Historical 1.00 default.xml

Eden_Historical	File name for the parameters for Eden_Historical.
1.00	Version number
default	Flag to indicate the version is the default configuration (otherwise omitted).

The structure of the module parameter XML configuration is the same as that applied in the Published Interface format. See the relevant documentation for the definition of the schema and required configuration.

### Template Module Parameter File

Since 2012.01 it is possible to add location attributes to the parameter file, using a template. Below is an example of a template.

Module parameter template for individual model-parameters

```
<parameters version="1.5" xmlns="http://www.wldelft.nl/fews/PI" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.wldelft.nl/fews/PI https://fewsdocs.deltares.nl/schemas/version1.0/pi-schemas/pi_modelparameters.xsd">
  <group id="MATCHES" name="MATCHES">
    <model>ExportParams</model>
    <parameter id="MATCH_ID" name="MATCH_ID">
      <description>Station used for matching</description>
      <stringValue>@EXTLOCID@</stringValue>
    </parameter>
  </group>
</parameters>
```

Use the templateLocationLoop-locationModelLoop functionality within the GeneralAdapter you can populate the file, either with a specific identified location, or by looping over a locationset to create a group per location. The <model> element is the key to connect the group-template to the looping functionality.

templateLocationLooping for a specific model

```
<exportParameterActivity>
  <fileName>OutputParameters</fileName>
  <templateLocationLooping>
    <locationModelLoop>
      <locationId>3201</locationId>
      <model>ExportParams</model>
    </locationModelLoop>
    <locationModelLoop>
      <locationSetId>URBS_models</locationSetId>
      <model>ExportParams</model>
    </locationModelLoop>
  </templateLocationLooping>
  <moduleInstanceId>ParametersTemplate</moduleInstanceId>
</exportParameterActivity>
```

A similar expansion loop-concept is available for to populate a parameter table, where the <parameter> element is the match between GA-loop and the template file

### Module parameter template for parameter table

```
<group id="@GROUPID@" name="@NAMEID@">
  <locationId>@PROJECTID@</locationId>
  < parameter id="SpillPriorityTable">
    <table>
      < columnIds A="SP_LOC" B="SPIL_DAY" C="SPIL_NIGHT"/>
      < columnTypes A="string" B="double" C="double"/>
      < row A="@SP_LOC@" B="@SP_DAY@" C="@SP_NIGHT@" />
    </table>
  < /parameter>
< /group>
```

Associated general adapter templateLocationLooping to create a group with a table per location

```
<templateLocationLooping>
  <locationTableLoop>
    <locationSetId>SpillPriorityLocations</locationSetId>
    <parameterId>SpillPriorityTable</parameterId>
  </locationTableLoop>
</templateLocationLooping>
```