# 22-3 Import from Deltares Open Archive

## Ingest datasets in Delft-FEWS stand alone - ImportArchiveModule

Once downloaded, the data can be ingested by a Delft-FEWS stand alone with the aim to bring the local datastore back in the 'original state', i.e. with a similar time series set definition as used when Delft-FEWS ran the workflow to produce the data. This exact timeseries definition is included in the netcdf files during the archive export process. The importArchiveModule has been implemented to uses this definition included in the netcdf files to put the data back in Delft-FEWS local datastore

Figure 4.16 and Table 4.10 describe the configuration details for the import module.
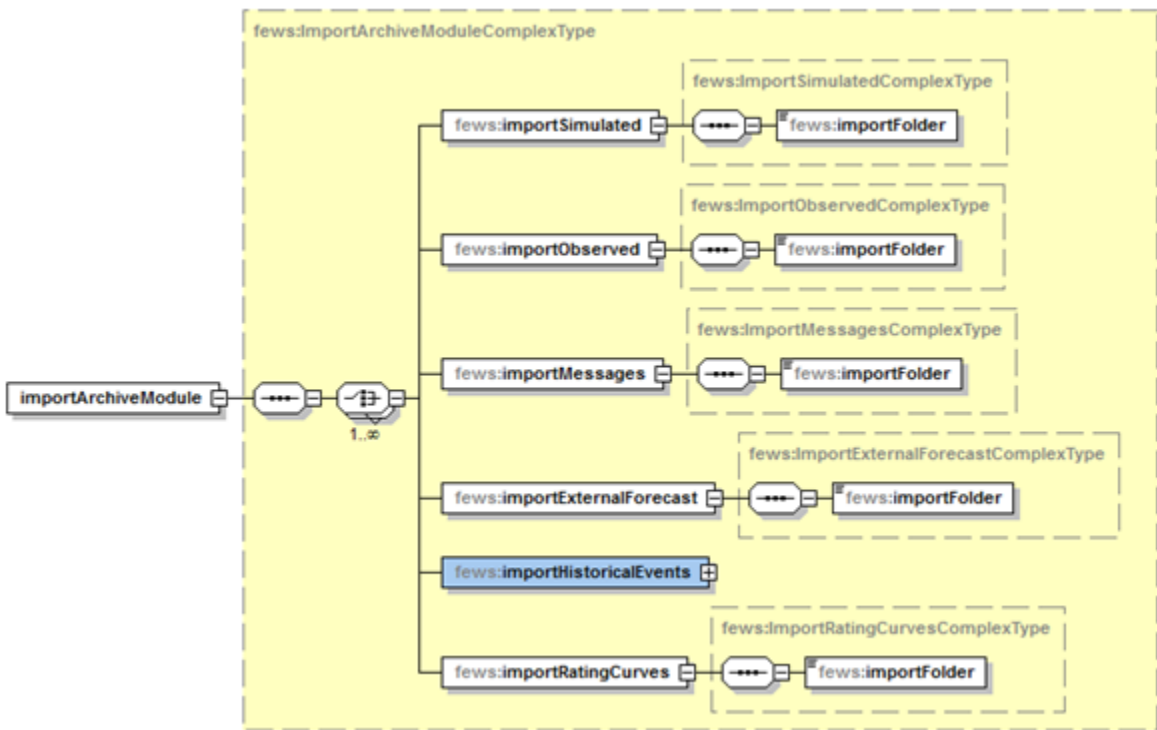


Figure 4.16 Configuration of archive Import (importArchiveModule.xsd)

Table 4.10 importArchiveModule configuration

| Element | Format | Description |
|---|---|---|
| importSimulated/importFolder | string (path) | Full path where the simulated datasets are made available for import by Delft-FEWS |
| importObserved/importFolder | string (path) | Full path where the observed datasets are made available for import by Delft-FEWS |
| importMessages/importFolder | string (path) | Full path where the messages datasets are made available for import by Delft-FEWS |
| importExternalForecast/importFolder | string (path) | Full path where the external forecasts are made available for import by Delft-FEWS |
| importRatingCurves/importFolder | string (path) | Full path where the rating curve datasets are made available for import by Delft-FEWS |

## IdMapping

By default it is **not** necessary to configure an idMap for the archive import. The variables containing time series in the netcdf-files in the open archive have an attribute called timeseries_sets_xml. This attribute contains a xml with the original timeseries definition. When the netcdf file is imported the time series in the variable will be mapped to the definition defined here. It is possible to define an idMapping for the archive import. This id mapping can only be used for mapping location ids and parameters ids. Note that the location id or parameter id defined in the xml doesn't have to be available anymore in FEWS.

## Ingest of Historic events in Delft-FEWS (FSS)

Historic events are a special datatype to Delft-FEWS as they can be used to overlay on an existing time series graph. Historic events are the only datasets which can be ingested in a Delft-FEWS client server system. Normally, the archive server has a process running (the HistoricEventsExporter) which extracts historic events data from the archive and pushes them to the Forecasting Shell Server import directories for ingest in the operational database.

This import process has its own data administration process, and does not use the timeseries definition as embedded in the netcdf files. Hence an idMap may be needed to translate netcdf variables into Delft-FEWS parameters and locations. In addition, backup and failure folders can be defined to prevent loss of data in the automated process.

Figure 4.17 and Table 4.11 describe the configuration for importing historic events. This module should be executed in a separate workflow compared to the other archived datasets.
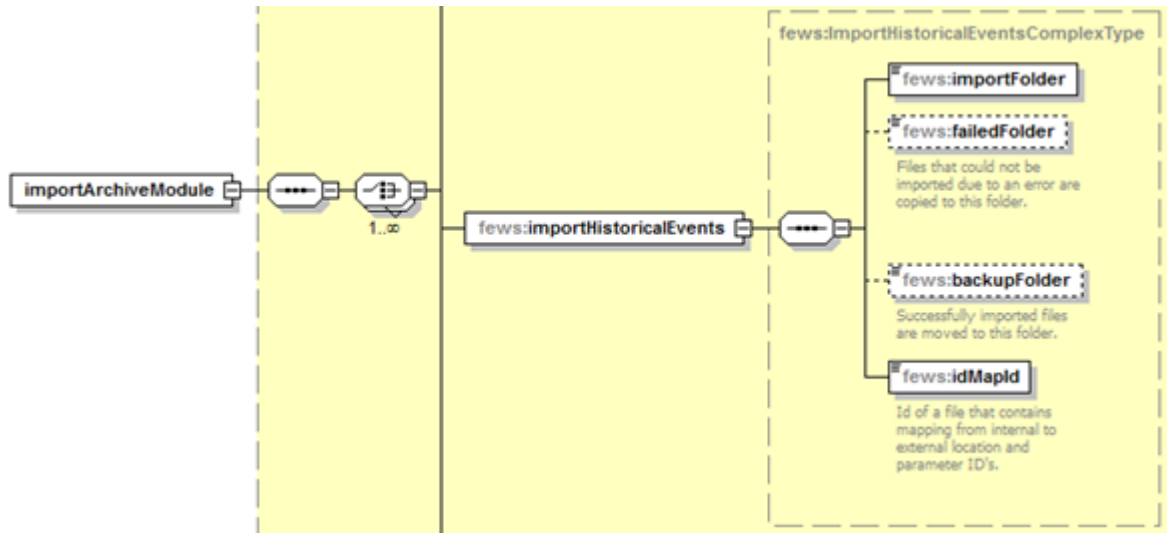


Figure 4.17 Configuration of historical events import (importArchiveModule.xsd)

Table 4.11 Historical events import configuration (importArchiveModule)

| Element | Format | Description |
| --- | --- | --- |
| importHistoricalEvents | importHistoricalEvents ComplexType | root element for import of Historical events into Delft-FEWS database |
| **importHistoricalEvents ComplexType** | | |
| importFolder | string (path) | Full path where the historical events  datasets are made available for import by Delft-FEWS |
| failedFolder | string (path | Full path where Delft-FEWS can put any dataset which failed on import |
| backupFolder | string (path | Full path where Delft-FEWS can back up any dataset which is being imported |
| idMapId | string | Idmap to translate NetCDF variables to FEWS parameters/locations |

## Import data from the Open Archive in a workflow

It is possible to schedule a workflow which imports data from the archive based on a list of data sets which is provided in a file. Below an example of the configuration of such a workflow.

```
<importArchiveModule xsi:schemaLocation="http://www.wldelft.nl/fews http://fews.wldelft.nl/schemas/version1.0
/importArchiveModule.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                                        xmlns="http://www.wldelft.nl/fews">
        <importRequestedDataSets>
                <importFolder>$IMPORT_FOLDER$</importFolder>
                <dataFolderArchive>$ARCHIVE_DOWNLOAD_FOLDER$</dataFolderArchive>
        </importRequestedDataSets>
</importArchiveModule>
```

The importFolder tag should define the directory which contains the files which define which datasets should be imported. The import folder is by default always the subfolder dataImport of the root of the archive.

The dataFolderArchive tag defines the location of the root of the data folder of the archive.

Below an example of how the files which define which datasets should be imported should look like.

```
<?xml version="1.0" encoding="UTF-8"?>
<dataSets xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.wldelft.nl/fews/archive" xsi:
schemaLocation="http://www.wldelft.nl/fews/archive http://fews.wldelft.nl/schemas//version1.0/archive-schemas
/dataImport.xsd">
    <dataSet>dataSet1\metaData.xml</dataSet>
    <dataSet>dataSet2\metaData.xml</dataSet>
</dataSets>
```

In the Open Archive each dataset has its own metaData.xml file. The import task will import each dataset which is defined in the list entirely. The import folder can contain multiple files. If that is the case the files will be imported one by one in a random order.

The files can be provided by writing them directly into the defined import folder. Another option is to upload the files. For this a dedicated webservice is available in the archive.

The webservice is available at the following relative URL at the archive web application /dataSetImportRequest/upload/. The file can be uploaded by using a post request.