

# Archives.xml - 2018.02 and later

- [Basic configuration](#)
  - [Test of the Archive configuration](#)
- [Optional configuration](#)
  - [ElasticSearchUrl](#)
  - [Reverse proxy](#)
  - [Archive database](#)
  - [Optional binDir](#)
  - [Uploading edited data to the archive](#)
  - [Custom export folders](#)

## Basic configuration

The Archives.xml can be used to configure the connection between your sa,oc or fss with the archive. For an sa or oc the configuration only consists of the catalogueUrl in case you have configured the archive according the standard (which is usually the case).

An example is given below.

```
<?xml version="1.0" encoding="UTF-8"?>
<archives xmlns="http://www.wldelft.nl/fews" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:
schemaLocation="http://www.wldelft.nl/fews https://fewsdocs.deltares.nl/schemas/version1.0/archives.xsd">
  <catalogueUrl>http://dummy_server:8080/deltares-archive-server/catalogue</catalogueUrl>
</archives>
```

Data from the Open Delft-FEWS Archive can be used in a Stand Alone or Operator Client FEWS client; this functionality is called seamless integration. The Archives.xml file, located in the SystemConfigFiles folder, is used to add this seamless integration functionality to the Delft-FEWS system. Seamless integration between the Open Delft-FEWS Archive and the Operational Delft-FEWS Database means that when data is not found in the operational database, FEWS will search for this data in the Open Archive.

The seamless integration functionality is now available in the Time Series Display for external historical scalar time series. Additional functionality for simulations and external historical time series is included in the FEWS pi-services.

Below an example of the content an Archives.xml configuration file:

```
<?xml version="1.0" encoding="UTF-8"?>
<archives xmlns="http://www.wldelft.nl/fews" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:
schemaLocation="http://www.wldelft.nl/fews https://fewsdocs.deltares.nl/schemas/version1.0/archives.xsd">
  <catalogueUrl>http://dummy_server:7080/deltares-archive-server/catalogue</catalogueUrl>
  <fileServerUrl>http://dummy_server:8081/thredds/fileServer/data</fileServerUrl>
  <dataFolder>d:\fews\archive\data</dataFolder>
</archives>
```

Note that a very common mistake made in configuring the Archives.xml is that the catalogueURL does not end with catalogue!

The only configuration needed in the Archives.xml is the location of the catalogue and thredds:

- The catalogue used for seamless integration is the internal elastic search catalogue (not the Geonetwork catalogue). The catalogue is used by the FEWS client to find archived data.
- The fileserver used for seamless integration in the THREDDS server. The fileserver is used to download the data from the archive.

The datafolder is optional. It is used by the pi-service (rest and soap) to read the data. If the datafolder is configured the data is read from the folder directly and not from THREDDS. Reading the data directly from disk is faster than reading the data from THREDDS. So the datafolder should be the root of the datafolder of the archive as it is on the server on which the pi-service is configured.

## Test of the Archive configuration

To test if the catalogue url is correct, paste the url in an internet browser. If the config is correct the response below will be visible (or something similar). The response either indicates that the catalogue is moved to another URL (releases prior to august 2020) or the response will look like

This url was used to host the embedded catalogue prior to the 201802 release.  
Now it is used to verify that the correct catalogueURL is configured correctly in the Archives.xml  
The URL `http://localhost:7080/deltares-archive-server/catalogue` should be used as the catalogueURL in the Archives.xml

To test if the fileServer url is correct append `/catalog.html` to the url configured and paste this url in an internet browser. If the url is correct a list of the files and directories in the root of the data folder in the archive will be visible.

## Optional configuration

### ElasticSearchUrl

By default FEWS will assume that the elastic catalogue is running at the same server as the archive web application at the port 9200.

The default can be changed by using the tag `elasticSearchUrl`. Below an example

```
<?xml version="1.0" encoding="UTF-8"?>
<archives xmlns="http://www.wldelft.nl/fews" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:
schemaLocation="http://www.wldelft.nl/fews https://fewsdocs.deltares.nl/schemas/version1.0/archives.xsd">
  <catalogueUrl>http://dummy_server:7080/deltares-archive-server/catalogue</catalogueUrl>
  <elasticSearchUrl>http://dummy_server:9800</elasticSearchUrl>
</archives>
```

### Reverse proxy

A FEWS OC or SA needs access to the catalogue to search for data in the archive. This requires *direct access to elastic* from a FEWS OC or SA, introducing a attack vector. Since August 2020 the Deltares Open Archive contains a *read-only reverse proxy* inside the Archive web application. This reverse proxy redirects only the limited set of read-only elastic requests which an FEWS OC or SA needs. The reverse proxy is available at the relative URL `/elastic` in the Archive web application. In order to make Operator Clients connect to the reverse proxy instead of connecting directly to elastic you should configure tag `elasticSearchUrl` like example below.

```
<?xml version="1.0" encoding="UTF-8"?>
<archives xmlns="http://www.wldelft.nl/fews" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:
schemaLocation="http://www.wldelft.nl/fews https://fewsdocs.deltares.nl/schemas/version1.0/archives.xsd">
  <catalogueUrl>http://dummy_server:7080/deltares-archive-server/catalogue</catalogueUrl>
  <elasticSearchUrl>http://dummy_server:7080/deltares-archive-server/elastic</elasticSearchUrl>
</archives>
```



Both the `catalogueUrl` and the `elasticSearchUrl` tags should be configured!

You can test the `elasticSearchUrl` by pasting the URL in a browser.

You should get a response like this.

```
{
  "name" : "gVb-boq",
  "cluster_name" : "elasticsearch",
  "cluster_uuid" : "LGGOTfhBRXyxxNDEX5CRuA",
  "version" : {
    "number" : "6.4.2",
    "build_flavor" : "oss",
    "build_type" : "zip",
    "build_hash" : "04711c2",
    "build_date" : "2018-09-26T13:34:09.098244Z",
    "build_snapshot" : false,
    "lucene_version" : "7.4.0",
    "minimum_wire_compatibility_version" : "5.6.0",
    "minimum_index_compatibility_version" : "5.0.0"
  },
  "tagline" : "You Know, for Search"
}
```

## Archive database

It is possible to store scalar data inside a database instead of in netcdf files. In the Archives.xml the details of the database (URL, username and password) and the details about how to load the plugin for the archive database should be configured. At the moment there is a plugin available for MongoDB.

A configuration example for the MongoDB plugin is available below:

```
<?xml version="1.0" encoding="UTF-8"?>
<archives 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/archives.xsd">
  <catalogueUrl>https://dummy_server/deltares-archive-server/</catalogueUrl>
  <elasticSearchUrl>https://dummy_server/deltares-archive-server/elastic</elasticSearchUrl>
  <archiveDatabase>
    <archiveDatabaseUrl>mongodb://server:port/fewsarchive?authSource=admin</archiveDatabaseUrl>
    <user>myuser</user>
    <password>mypass</password>
    <pluginClass>nl.fews.archivedatabase.mongodb.MongoDbArchiveDatabase</pluginClass>
    <pluginFactoryMethod>create</pluginFactoryMethod>
    <openDatabasePluginPanelBinDir>d:\mongodb-plugin\</openDatabasePluginPanelBinDir>
  </archiveDatabase>
</archives>
```

The archiveDatabaseUrl and the username and password are dependent on the system setup. If you use the MongoDB-plugin then you should use the pluginClass and pluginFactoryMethod as shown in the example. The plugin can be provided by Deltares on request.

## Optional binDir

You can configure a custom directory to install the jar-files of the plugin in. If you do not configure a custom pluginDir then you should located the plugin . jar file in the same directory as the FEWS jars.

## Uploading edited data to the archive

When data is edited in FEWS then the data edits are by default not send to the archive. When the option uploadEditedDataToOpenArchive is set the data edits are send to the Archive.

It also necessary to configure the uploadEditedDataToOpenArchive. This location attribute function will be used to determine to which area (one or more) each location belongs.

Below a configuration example

```
<?xml version="1.0" encoding="UTF-8"?>
<archives 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/archives.xsd">
  <catalogueUrl>http://localhost:7080/deltares-archive-server/catalogue</catalogueUrl>
  <elasticSearchUrl>http://localhost:7080/deltares-archive-server/elastic</elasticSearchUrl>
  <areaLocationAttributeFunction>@station:REPORT_FOLDER@</areaLocationAttributeFunction>
  <uploadEditedDataToOpenArchive>true</uploadEditedDataToOpenArchive>
</archives>
```

In addition a workflow needs to be scheduled to insert the data edits into the archive. More details can be found here [22-4 Merging edited data - DELFT-FEWS Documentation - Deltares Public Wiki](#)

## Custom export folders

Data is exported to the open archive according to fixed folder structure. In some cases this folder structure is customizable by using configuration options. From FEWS Release 202302 and higher it is possible to configure a custom export folder for each data type in the archive.

A configuration example is given below:

```
<?xml version="1.0" encoding="UTF-8"?>
<archives xmlns="http://www.wldelft.nl/fews" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:
schemaLocation="http://www.wldelft.nl/fews https://fewsdocs.deltares.nl/schemas/version1.0/archives.xsd">
  <catalogueUrl>https://fews-ut-ar-202301.avi.deltares.nl/deltares-archive-server/catalogue</catalogueUrl>
  <elasticSearchUrl>https://fews-ut-ar-202301.avi.deltares.nl/deltares-archive-server/elastic</elasticSearchUrl>
  <dataFolder>C:\fews\archive\data</dataFolder>
  <customExportFolders>
    <products>yyyy/MM'/%areaId%/sourceId/'dd'/products/%productId%_currentTime%</products>
    <config>yyyy/MM'/%areaId%/revisionId%</config>
    <forecasterNotes>yyyy/MM'/%areaId%/test/'dd'/forecasterNotes</forecasterNotes>
    <logMessages>yyyy/MM/dd'/%areaId%/logmessages</logMessages>
    <ratingCurves>yyyy/MM/dd'/%areaId%/ratingcurves</ratingCurves>
    <snapshot>yyyy/MM/dd'/%areaId%/snapshot</snapshot>
    <externalHistorical>yyyy/MM'/%areaId%/sourceId/'dd'/observed_%valueType%</externalHistorical>
    <externalForecast>yyyy/MM'/%areaId%/sourceId/'dd'/externalForecast/%sourceId%_HHmmss</externalForecast>
    <simulatedForecast>yyyy'/%taskRunId%/areaId%/sourceId%/whatIfId%/workflowId%_forecastTime%_currentTime%</simulatedForecast>
    <simulatedHistorical>yyyy/MM'/%areaId%/sourceId/'dd'/simulatedHistorical_%valueType%</simulatedHistorical>
  </customExportFolders>
</archives>
```

It is possible to use a date/time pattern in the folder structure. The pattern will be parsed during export by using the time relevant for the export. The time that is used to parse the date/time pattern is dependent on the type of export. Below an overview of the time used for each export type.

	Time used
External historical	start time exported data
External forecast	external forecast time
Simulated forecast	time zero
Simulated historical	start time exported data
Snapshot	current time
Config	creation time latest revision
Products	time zero
Rating curves	current time
Log messages	start time exported data
Forecaster notes	start time exported data

In addition it is possible to use plain text in the folder structure.

A set of tags like %areald% are supported and will be parsed during the export. Note, however, that not all tags are supported for each type of export; below overview shows which tags are supported for each export type.

	areald	sourceld	currentTime	taskRunId	whatIfId	workflowId	supplied attributes	revisionId	
External historical	x	x							
External forecast	x	x							
Simulated Forecast	x	x	x	x	x	x			
Simulated historical	x								
Snapshot	x								
Config	x							x	
Products	x	x	x				x		
Rating curves	x								
Log messages	x								
Forecaster notes	x								