

External netcdf storage

What is the external netcdf storage?

The external netcdf storage was added to the Deltares Open Archive in 2021. It can be used to store netcdf files in your Open Archive.

These netcdf-files are typically not produced by your own FEWS system but by another system. The main constraint is that the netcdf files are CF compliant.

The external netcdf storage facilitates storing external historical and external forecasts data. Both scalar and grid data are supported. It is possible to store the actual data in your external storage but it is also possible to reference an externally hosted THREDDS netcdf server.

How should a netcdf be formatted

- all variables should have the following attributes: "standard_name", "long_name" and "units".

Example Python script to make netcdf file according to requirements

How can I use the data in the external netcdf storage?

Once you have made the data available it is possible to retrieve the time series from the external netcdf storage by using the FEWS webservices (WMS or pi timeseries). It is possible to retrieve external historical and external forecast data from the netcdf storage.

In addition the data is available in the TSD or Grid display by using the seamless integration. This option is only supported for external historical data. If you have enabled seamless integration for you workflow then the workflow will import external historical data which is available in the external netcdf storage but not (yet) in FEWS before running the workflow.

The archive display can be used to download and import data from the external netcdf storage. It is also possible to define events for the external netcdf storage.

How can I configure an external netcdf storage?

As mentioned earlier the external netcdf storage is only supported for CF compliant netcdf files. Below we will start with an example on how to configure an external netcdf storage.

It is possible to define 1 or more external netcdf storages for your archive. The external storages should be defined in the archive itself in the ExternalStorages.xml file. In addition the external netcdf storages should be defined in the Archives.xml file of your FEWS system.

We will start with explaining how the external netcdf storages should be defined in the archive itself.

ExternalStorages.xml

This file should be located in the config folder of your archive.

You can place the file directly in this folder but you can also upload this file using the manage configuration tab in the archive web gui.

The example below shows the minimal configuration that is needed to define an external netcdf storage.

```
<?xml version="1.0" encoding="UTF-8"?>
<externalStorage xsi:schemaLocation="http://www.wldelft.nl/fews/archive http://fews.wldelft.nl/schemas
//version1.0/archive-schemas/externalStorage.xsd" xmlns="http://www.wldelft.nl/fews/archive" xmlns:xsi="
http://www.w3.org/2001/XMLSchema-instance">
  <externalCFCompliantNetCDFStorage id="Maps1D_Rivieren">
    <dataFolder>D:\FEWS\archive\matroos\maps1d</dataFolder>
    <timeSeriesType>externalForecast</timeSeriesType>
    <attributeMapping>
      <moduleInstanceId>system</moduleInstanceId>
    </attributeMapping>
  </externalCFCompliantNetCDFStorage>
</externalStorage>
```

The data folder defines the root directory in which all the netcdf files are stored. Note that it is allowed to use subfolders. The tag timeSeriesType defines which type of data (external forecast or external historical) is stored in this folder.

Timeseries mapping

To map the timeseries in the netcdf file to a FEWS timeseries definition attributes are used. In the attributeMapping can be defined which attribute will be used to define the module instance id of the timeseries in the netcdf file.

It is always mandatory to define which attributes will define the module instance id. The location id and the parameter id is defined in the netcdf-files itself. It is therefore not needed to define a attribute for the location id or the parameter id.

If you only define an attribute mapping for a module instance here then the following defaults will be assumed. The ensemble id is main, the time series has no qualifiers, the time step is non equidistant and the data is scalar data.

If you want to define which attributes define the time step, qualifiers and the ensemble id then this should be added to the attributeMapping. Below an example.

```
<?xml version="1.0" encoding="UTF-8"?>
<externalStorage xsi:schemaLocation="http://www.wldelft.nl/fews/archive http://fews.wldelft.nl/schemas
//version1.0/archive-schemas/externalStorage.xsd" xmlns="http://www.wldelft.nl/fews/archive" xmlns:xsi="
http://www.w3.org/2001/XMLSchema-instance">
  <externalCFCompliantNetCDFStorage id="Maps1D_Rivieren">
    <dataFolder>D:\FEWS\archive\matroos\maps1d</dataFolder>
    <timeSeriesType>externalForecast</timeSeriesType>
    <attributeMapping>
      <moduleInstanceId>system</moduleInstanceId>
      <timeStepId>timeStepAttribute</timeStepId>
      <qualifiersId>qualifierAttribute</qualifiersId>
      <ensembleId>ensembleAttribute</ensembleId>
    </attributeMapping>
  </externalCFCompliantNetCDFStorage>
</externalStorage>
```

If you are using netcdf-files in which these attributes are not available then you can configure attribute values in the xml-file itself. Below an example.

```
<?xml version="1.0" encoding="UTF-8"?>
<externalStorage xsi:schemaLocation="http://www.wldelft.nl/fews/archive http://fews.wldelft.nl/schemas
//version1.0/archive-schemas/externalStorage.xsd" xmlns="http://www.wldelft.nl/fews/archive" xmlns:xsi="
http://www.w3.org/2001/XMLSchema-instance">
  <externalCFCompliantNetCDFStorage id="Maps1D_Rivieren">
    <dataFolder>D:\FEWS\archive\matroos\maps1d</dataFolder>
    <attribute id="timeStepAttribute" name="SETS360"/>
    <timeSeriesType>externalForecast</timeSeriesType>
    <attributeMapping>
      <moduleInstanceId>system</moduleInstanceId>
      <timeStepId>timeStepAttribute</timeStepId>
      <qualifiersId>qualifierAttribute</qualifiersId>
      <ensembleId>ensembleAttribute</ensembleId>
    </attributeMapping>
  </externalCFCompliantNetCDFStorage>
</externalStorage>
```

Area id and source id

The archive display searches the archive by using an area id and a source id. Both are used to facilitate quick searching in your archive. It is possible to define which attributes of a netcdf file define the area id and source id.

They can also be defined in the attributeMapping section. Below an example.

```
<?xml version="1.0" encoding="UTF-8"?>
<externalStorage xsi:schemaLocation="http://www.wldelft.nl/fews/archive http://fews.wldelft.nl/schemas
//version1.0/archive-schemas/externalStorage.xsd" xmlns="http://www.wldelft.nl/fews/archive" xmlns:xsi="
http://www.w3.org/2001/XMLSchema-instance">
  <externalCFCompliantNetCDFStorage id="Maps1D_Rivieren">
    <dataFolder>D:\FEWS\archive\matroos\maps1d</dataFolder>
    <timeSeriesType>externalForecast</timeSeriesType>
    <attributeMapping>
      <moduleInstanceId>system</moduleInstanceId>
      <timeStepId>SETS360</timeStepId>
      <qualifiersId>q1</qualifiersId>
      <ensembleId>era5</ensembleId>
      <areaId>myAreaId</areaId>
      <sourceId>mySourceId</sourceId>
    </attributeMapping>
  </externalCFCompliantNetCDFStorage>
</externalStorage>
```

Grid id

In FEWS a grid has a location id. However such a definition is not available for grids in Netcdf. The location id must therefore be defined by an attribute to facilitate that a time series in a netcdf file can be mapped to a time series in FEWS.

This can be done by defining an attribute which contains the location id for the grid. Below an example.

```
<?xml version="1.0" encoding="UTF-8"?>
<externalStorage xsi:schemaLocation="http://www.wldelft.nl/fews/archive http://fews.wldelft.nl/schemas
//version1.0/archive-schemas/externalStorage.xsd" xmlns="http://www.wldelft.nl/fews/archive" xmlns:xsi="
http://www.w3.org/2001/XMLSchema-instance">
  <externalCFCompliantNetCDFStorage id="Maps1D_Rivieren">
    <dataFolder>D:\FEWS\archive\matroos\maps1d</dataFolder>
    <timeSeriesType>externalForecast</timeSeriesType>
    <attributeMapping>
      <moduleInstanceId>system</moduleInstanceId>
      <timeStepId>SETS360</timeStepId>
      <qualifiersId>q1</qualifiersId>
      <gridId>locationForGrid</gridId>
      <ensembleId>era5</ensembleId>
      <areaId>myAreaId</areaId>
      <sourceId>mySourceId</sourceId>
    </attributeMapping>
  </externalCFCompliantNetCDFStorage>
</externalStorage>
```

Data folder and file name filter

Usually a data folder for the external netcdf storage contains a lot of different data. It is usually inconvenient or even impossible to assign all the data to a single netcdf storage.

If for example a data folder contains scalar data and grid data then the data cannot be assigned to a single external storage because a single external storage always contains only scalar or only grid data.

It is therefore common to assign only a part of the data to a single netcdf storage. To define which netcdf file should be assigned to a specific netcdf storage you can use data folder filters and file name filters

They can be used to define which folders and which files belong to the netcdf storage. Below an example.

```

<?xml version="1.0" encoding="UTF-8"?>
<externalStorage xsi:schemaLocation="http://www.wldelft.nl/fews/archive http://fews.wldelft.nl/schemas
//version1.0/archive-schemas/externalStorage.xsd" xmlns="http://www.wldelft.nl/fews/archive" xmlns:xsi="
http://www.w3.org/2001/XMLSchema-instance">
  <externalCFCompliantNetCDFStorage id="Maps1D_Rivieren">
    <dataFolder>D:\FEWS\archive\matroos\maps1d</dataFolder>
    <timeSeriesType>externalForecast</timeSeriesType>
    <dataFolderFilter>
      <allValid>
        <dataFolderContains contains="areaId"/>
      </allValid>
      <anyValid>
        <dataFolderContains contains="abc"/>
        <dataFolderContains contains="def"/>
      </anyValid>
      <not>
        <anyValid>
          <dataFolderContains contains="123"/>
        </anyValid>
      </not>
    </dataFolderFilter>
    <fileNameFilter>
      <allValid>
        <fileContains contains="name"/>
      </allValid>
      <anyValid>
        <fileContains contains="123"/>
      </anyValid>
      <not>
        <anyValid>
          <fileContains contains="123"/>
        </anyValid>
      </not>
    </fileNameFilter>
    <attributeMapping>
      <moduleId>system</moduleId>
      <timeStepId>SETS360</timeStepId>
      <qualifiersId>q1</qualifiersId>
      <gridId>locationForGrid</gridId>
      <ensembleId>era5</ensembleId>
      <areaId>myAreaId</areaId>
      <sourceId>mySourceId</sourceId>
    </attributeMapping>
  </externalCFCompliantNetCDFStorage>
</externalStorage>

```

Reference an external THREDDS server

It is possible to include an externally hosted THREDDS server into your OpenArchive. Instead of an data folder you should then define the URL of the the THREDS catalogue and URL of the THREDS file server.

Below an example.

```
<?xml version="1.0" encoding="UTF-8"?>
<externalStorage xsi:schemaLocation="http://www.wldelft.nl/fews/archive http://fews.wldelft.nl/schemas
//version1.0/archive-schemas/externalStorage.xsd" xmlns="http://www.wldelft.nl/fews/archive" xmlns:xsi="
http://www.w3.org/2001/XMLSchema-instance">
  <externalCFCompliantNetCDFStorage id="observedscalarstorage">
    <thredsCatalogueURL>https://fews-ut-ar-202201.avi.deltares.nl/thredds/catalog/data/netcdf-
storage/observedscalarstorage/catalog.xml</thredsCatalogueURL>
    <threddsFileServerURL>https://fews-ut-ar-202201.avi.deltares.nl/thredds/fileServer/<
/threddsFileServerURL>
    <timeSeriesType>observed</timeSeriesType>
    <attributeMapping>
      <moduleInstanceId>attributeForModuleInstance</moduleInstanceId>
      <areaId>areaId</areaId>
      <sourceId>sourceId</sourceId>
    </attributeMapping>
  </externalCFCompliantNetCDFStorage>
</externalStorage>
```

layerLocationPrefix

For z-layer or sigma layer time series a location id will be generated for each layer by using the defined prefix. If a prefix layer is defined then the layer with id 3 will have location id layer3.

If this option is not defined then each layer will have location id equal to the layer index.

attribute

With this option it is possible to add attribute key-value pairs to a external netcdf storage. They will be added to the attributes defined in the netcdf file.

Simulated netcdf storage

Simulations which are stored in the Open Archive consists of several parts:

- time series (netcdf)
- reports
- states
- modifiers (xml)
- what-ifs (xml)

In the Open Archive all these parts are stored in sub folders of a single directory.

It is possible to store the time series in the netcdf storage and the other parts in the Open Archive . This can be configured in the archive export for simulated data in FEWS by using the tag <netcdfStorageExport>

To make these time series in these netcdf files available you need to configure a simulatedNetcdfForecastingNetcdfStorage in your ExternalStorages.xml. Each exported netcdf file will contain a task run id. The task run id will be used to match

the netcdf-files to the simulations in the Open Archive. When you export multiple simulations to the Open Archive during a single run in FEWS then it is not possible anymore to know to which simulation a netcdf-file belongs because there will be multiple simulations in the archive with the same task run id. For this situation the matchingAttributeId should be used. In this case the netcdf files will also be matched by using the values of the defined matching attributes.

```
<?xml version="1.0" encoding="UTF-8"?>
<externalStorage xsi:schemaLocation="http://www.wldelft.nl/fews/archive http://fews.wldelft.nl/schemas
//version1.0/archive-schemas/externalStorage.xsd" xmlns="http://www.wldelft.nl/fews/archive" xmlns:xsi="
http://www.w3.org/2001/XMLSchema-instance">
  <simulatedForecastingNetcdfStorage id="simulated">
    <dataFolder>D:\FEWS\archive\matroos\maps1d</dataFolder>
    <matchingAttributeId>attributeA</matchingAttributeId>
    <matchingAttributeId>attributeB</matchingAttributeId>
  </simulatedForecastingNetcdfStorage>
</externalStorage>
```

Archives.xml

Besides defining the external netcdf storages at the archive itself they also need to be defined in FEWS by using the Archives.xml.

Note that the id's used in the Archives.xml and the ExternalStorages.xml need to match exactly.

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 http://fews.wldelft.nl/schemas/version1.0/archives.xsd">
  <catalogueUrl>http://localhost:7080/deltares-archive-server</catalogueUrl>
  <externalCFCompliantNetCdfStorage id="Maps1D_Noordzee_ext_historical">
    <idMapId>IdImportArchive_Waqua_dcsmv6_harmonie</idMapId>
    <openDapUrl>http://localhost:7080/thredds/dodsC/matroos/maps1d</openDapUrl>
    <fileServerUrl>http://localhost:7080/thredds/dodsC/matroos/maps1d</fileServerUrl>
  </externalCFCompliantNetCdfStorage>
</archives>
```

The IdMapId defines the id map for the external netcdf storage. The OpenDapUrl defines the url which is used for the seamless integration. The url defined here plus the relative path in the netcdf storage should define the url by which the time series can be read using OpenDAP.

The fileServerUrl defines the url by which the raw netcdf file can be downloaded. The url defined here plus the relative path in the netcdf storage should define the url by which the netcdf file can be downloaded.