NetCDF formats that can be imported in Delft-FEWS

Contents

- Contents
- NetCDF formats in general
- NetCDF formats that can be imported in Delft-FEWS
- NetCDF kickstarter tool
- Checklist to create NetCDF data in the format of the Netcdf CF-1.6 conventions
    - All data
    - Gridded data
    - Compacted data
- How to import a netCDF file into Delft-FEWS
    - Import configuration
    - Creating the correct grid definition
- Error messages from the Delft-FEWS NetCDF import
    - Invalid URL
    - Errors from the Java NetCDF library
- Related modules and documentation
    - Internal
    - External

NetCDF formats in general

The NetCDF file format is very flexible in the ways that it allows the data to be stored in the file. Unfortunately this means that NetCDF files come in many different formats and the way the variables and dimensions (data structure) are organised in the file can be completely different for different NetCDF files. Therefore it cannot be expected that the FEWS NetCDF import will be able to import every single netcdf file that's out there. Because it is just not possible to anticipate all the possible data structures that people can come up with.

For this reason there exist several conventions that describe standard ways of organising the data in NetCDF files. The FEWS NetCDF import is based on the CF-1.6 conventions. This means that if a NetCDF file is in a format that is fully compatible with the CF-1.6 conventions, then there is a good chance that FEWS will be able to import the data from the NetCDF file. However there are a few exceptions, which are listed on this page. For instance FEWS can only import data that depends on a time dimension (time series), while the CF-1.6 conventions also describe formats for data that does not depend on time. In other words: Delft-FEWS restricts the NetCDF CF-1.6 conventions by applying extra limitations on the format of the NetCDF files.

Furthermore there are some exceptions where FEWS can import NetCDF data, even though it does not comply with the CF-1.6 conventions. This can be due to historic reasons. Or this can be due to the fact that some data formats are not (yet) part of the CF-1.6 conventions (e.g. spectral data), in which case these exceptions are also described on this page.

NetCDF formats that can be imported in Delft-FEWS

⚠ TODO work in progress

A first check to determine whether a given NetCDF file can be imported in FEWS is to use an online CF compliance checker, see e.g. http://cfconventions.org/compliance-checker.html or https://compliance.ioos.us/index.html. If the NetCDF file is not valid according to the CF compliance checker, then the NetCDF file can probably not be imported in FEWS. In this case please contact the data supplier that created the NetCDF file and ask them to make the NetCDF file CF-1.6 compliant. If the file is not CF-1.6 compliant then there is no reason to contact the FEWS support team.

Also note that there can be slight differences between importing a grid from a netCDF file into Delft-FEWS and into other viewers. For example a netCDF file with a rotated pole grid that contains both one-dimensional coordinate variables in the rotated coordinate system and two-dimensional coordinate variables in the WGS1984 system. Panoply will read from this file the two-dimensional coordinate variables. Delft-FEWS can read from this file either the two-dimensional coordinate variables (irregular grid definition) or the one-dimensional coordinate variables (regular grid definition, i.e. more efficient, better performance). If the one-dimensional coordinate variables or grid mapping in this netCDF file are not CF compliant, then this might result in problems in Delft-FEWS, which are not visible in Panoply.

Scalar Time Series Example

Netcdf file: scalar_time_series.nc

Text version: scalar_time_series.txt

Regular Grid Time Series Example

Netcdf file: regular_grid_time_series.nc

Text version: regular_grid_time_series.txt

Curvilinear Grid Time Series Example

Netcdf file: curvilinear_grid_time_series.nc

Text version: curvilinear_grid_time_series.txt

Unstructured Grid (UGRID) Time Series

For an example of data in UGRID format, see http://ugrid-conventions.github.io/ugrid-conventions/

To import unstructured grid data in Delft-FEWS you need to add an irregular grid definition to the grids.xml configuration file for your data. For more information see 06 Grids#06Grids-UsingaUGRIDNetCDFfileasunstructuredgriddefinition

Track Time Series (single trajectory per file)

For featureType "point" all points in the file are imported together to form a single track. For featureType "trajectory" the parser supports only a single track per file, i.e. variables in the NetCDF file must be one dimensional and only have a time dimension. For an example of the track data format that can be imported, see http://cfconventions.org/cf-conventions/cf-conventions.html#_single_trajectory

NetCDF kickstarter tool

The netCDF kickstarter is a very useful tool for creating template netCDF files and is available on http://zandmotor.citg.tudelft.nl/netcdfkickstarter/ This tool can be used to create template netCDF files or to create scripts that generate template netCDF files. In this tool the user can specify what data variables the netCDF file should contain and the netCDF kickstarter will create a template netCDF file that contains the corresponding metadata variables and empty data variables, fully compliant according to the CF conventions. Then the user only needs to add the actual data values to this template file in order to create a CF compliant netCDF file. The kickstarter can also generate a script (in one of several programming languages) to create a template netCDF file that contains the metadata variables and empty data variables specified by the user. Then the user only needs to extend this script with commands to write the actual data values to the netCDF file.

Checklist to create NetCDF data in the format of the Netcdf CF-1.6 conventions

> ⚠ TODO work in progress

This checklist focusses on the points that are important to be able to import the netcdf data into Delft-FEWS. The full NetCDF CF-1.6 conventions can be found on http://cfconventions.org/Data/cf-conventions/cf-conventions-1.6/build/cf-conventions.html.

Before creating/adapting any code/script to produce netCDF files, it can be very useful to try the netCDF kickstarter tool described above (http://zandmotor.citg.tudelft.nl/netcdfkickstarter/), and/or read the following webpages:

- http://www.unidata.ucar.edu/publications/factsheets/current/netcdf_factsheet.pdf
- https://www.unidata.ucar.edu/software/netcdf/docs/BestPractices.html
- http://cfconventions.org/Data/cf-conventions/cf-conventions-1.6/build/cf-conventions.html

Furthermore some links with reference documentation about netCDF files:

- https://github.com/cf-convention/cf-documents/blob/master/other/cf_overview_viewgraphs.pdf?raw=true
- https://www.unidata.ucar.edu/software/netcdf/workshops/2012/cf/index.html
- https://www.unidata.ucar.edu/software/netcdf/workshops/2012/index.html

All data

The global attribute "Conventions" must be specified in the netcdf file (e.g. Conventions="CF-1.6"). If this attribute is not present or not valid, then the NetCDF Java library assumes a default convention (different from CF), which can cause unexpected behaviour in the NetCDF import in Delft-FEWS. Also see http://cfconventions.org/Data/cf-conventions/cf-conventions-1.6/build/cf-conventions.html#identification-of-conventions.

Gridded data

See for references to the cf-conventions website below.

1. all variables (data and coordinate) must have a unique variable name and the following attributes:
    - a. units (required, see units link below)
    - b. standard_name (required, see standard_name link below) - should comply with standard names list
    - c. long_name (recommended, see long_name links below)
    - d. _FillValue and/or missing_value (required if data contains missing values)
    - e. scale_factor (optional to compact data)
    - f. add_offset (optional to compact data)
2. Data variable for a 2D grid must depend on three dimensions, representing time and two spatial dimensions (in either the order time, y, x or time, x, y). The names of the dimensions can be anything.
    - a. Time dimension must have a corresponding 1D time coordinate variable with the same name.
    - b. Spatial dimensions, for
        - i. regular or rectangular grids - each spatial dimension must have a corresponding 1D coordinate variable with the same name (see coordinates link below).
        - ii. curvilinear grids - both spatial dimensions are 2D and depend on both spatial dimensions.
            1. The data variable must have an attribute "coordinates" that refers to the y and x coordinate variables (in that order), e.g. coordinates="y x"
            2. Each coordinate variable must have an attribute "axis" that indicates which axis is defined by that coordinate variable.
                - a. Time coordinate variable requires axis "T"
                - b. x coordinate variable requires axis "X"
                - c. y coordinate variable requires axis "Y"
                - d. (optional) analysis_time coordinate, see below

3. Time coordinate variable must have the attributes
    a. "units" with a valid unit of time - e.g. units="minutes since 1970-01-01 00:00:00.0 +0000" (time coordinate link below)
    b. "axis" with value "T"
4. Time and spatial coordinate variables must have values either in ascending or in descending order.
5. Forecast reference time/analysis time/base time (optional), must be stored in a separate coordinate variable that
    a. contains only one value: the forecast reference time, and
    b. standard_name="forecast_reference_time"

    The name of this variable must be present in the coordinates attribute of the data variable, e.g. coordinates="analysis_time" or coordinates="y x analysis_time".
    If needed, different data variables can point to different forecast reference time variables (see scalar coordinate link below).

Topic	Link to cfconventions.org
units	http://cfconventions.org/Data/cf-conventions/cf-conventions-1.6/build/cf-conventions.html#units
standard_name	http://cfconventions.org/Data/cf-conventions/cf-conventions-1.6/build/cf-conventions.html#standard-name http://cfconventions.org/Data/cf-standard-names/26/build/cf-standard-name-table.html
long_name	http://cfconventions.org/Data/cf-conventions/cf-conventions-1.6/build/cf-conventions.html#long-name
regular grids Lat, Long, Vert, and Time Axes	http://cfconventions.org/Data/cf-conventions/cf-conventions-1.6/build/cf-conventions.html#idp5553648
coordinates 2D Lat, Long, coordinate	http://cfconventions.org/Data/cf-conventions/cf-conventions-1.6/build/cf-conventions.html#idp5559280
time coordinate	http://cfconventions.org/Data/cf-conventions/cf-conventions-1.6/build/cf-conventions.html#time-coordinate
scalar coordinate	http://cfconventions.org/Data/cf-conventions/cf-conventions-1.6/build/cf-conventions.html#scalar-coordinate-variables

Compacted data

Netcf data variables with attributes scale_factor and add_offset will be treated as compacted data and the values of these attributes will be used to uncompact the values of this variable.

How to import a netCDF file into Delft-FEWS

To import data from a netCDF file into Delft-FEWS you must first create the relevant configuration.

Import configuration

To import data from a netCDF file into Delft-FEWS, either create a time series import run (see 03 Import Module) or an importNetcdfActivity in a general adapter (see 05 General Adapter Module#05GeneralAdapterModule-importNetcdfActivity). For an import run, use one of the following importTypes, depending on the type of data that you want to import:

NETCDF-CF_TIMESERIES, NETCDF-CF_GRID, NETCDF-CF_PROFILE or NETCDF-CF_TRAJECTORY.

Do not use the obsolete importType "NetcdfGridDataset". Only use importType "NetcdfGridDataset" for grid NetCDF files that contain ensembles or qualifiers.

To import scalar time series you also need to configure the corresponding internal locations (see 01 Locations) and an id mapping (see 01 ID Mapping).

To import grid time series you also need to configure the grid definition in Delft-FEWS, see the next section.

Creating the correct grid definition

For irregular or unstructured grids the cell center (and cell boundaries) must be defined for each grid cell. See 06 Grids#06Grids-Irregulargrids for ways to do this.

For a grid that is regular in a certain coordinate system, follow these steps (also see 06 Grids#06Grids-Regulargrids):

1. In the grids.xml config file create an empty regular grid definition.
2. Open the netCDF file in your favorite viewer and find the details of the coordinate system in which the grid is a regular grid. This coordinate system is usually described in a grid mapping variable, which has an attribute called "grid_mapping_name". In your grid definition choose the geoDatum or grid projection that corresponds to this coordinate system.
3. In the netCDF file find the variables that contain the coordinates of the grid cells *in the coordinate system in which the grid is regular*. These are one-dimensional variables that usually have standard names like "projection_x_coordinate" and "projection_y_coordinate" or "grid_longitude" and "grid_latitude". According to the CF conventions these coordinates are cell centers. Find the lowest x coordinate and use that as firstCellCenter x in your grid definition. Find the highest y coordinate and use that as firstCellCenter y in your grid definition.
4. In the netCDF file look at the spacings between the x and y coordinates. If these spacings are constant, then use them as xCellSize and yCellSize in your grid definition.
5. In the netCDF file find the sizes of the dimensions of the coordinate variables. Use these as the number of rows and columns in your grid definition.

If, after following these instructions carefully, the netCDF file is still not imported correctly into Delft-FEWS, then the file is probably not valid according to the NetCDF-CF conventions. In that case try to make the netCDF file CF compliant. As a last resort, you can try to use the obsolete option gridStartPoint (to swap the grid in the north-south direction) or the obsolete importType "NetcdfGridDataset", *but only as a last resort*.

Error messages from the Delft-FEWS NetCDF import

Invalid URL

If you get the error "invalid URL" then either the URL is invalid, or the OpenDAP server does not return any data. The latter either means that the OpenDAP server itself does not work correctly, or that the user does not have access rights to download the file on the OpenDAP server. This can be easily verified by following these steps:

1. copy the URL of the netcdf file or catalog from the FEWS configuration or from the error message.
2. if the URL ends with .nc, then append .html to it, so that it ends with .nc.html
3. go to the resulting URL in a browser, e.g. Mozilla Firefox.
4. if the browser gives an error, then there is a problem with the URL, the OpenDAP server or user access rights to the server. In this case please contact the administrator of the OpenDAP server (and not the FEWS support team).

Errors from the Java NetCDF library

If you get an error like e.g. "Failed to open NetCdf file from url" or "Variable name (...) must be unique within Group", then the netcdf file is probably corrupt or not CF-1.6 compliant. If the error message is accompanied by a stacktrace (in the FEWS log.txt file) then look for the first part of the stacktrace. Error messages from the Java NetCDF library can be recognised by a line that starts with something like "ucar.nc2.", e.g.:

```
java.lang.IllegalArgumentException: Variable name (hgtpv34768) must be unique within Group
version: 2014.02, build: 54174 (patched from 50595) May21, type: stable, jre: 1.7.0_40, os:Windows 7, mx: 1037m,
db: Firebird
            at ucar.nc2.Group.addVariable(Group.java:458)
            at ucar.nc2.dods.DODSNetcdfFile.addVariable(DODSNetcdfFile.java:646)
            at ucar.nc2.dods.DODSNetcdfFile.constructConstructors(DODSNetcdfFile.java:634)
            at ucar.nc2.dods.DODSNetcdfFile.<init>(DODSNetcdfFile.java:277)
            at ucar.nc2.dataset.NetcdfDataset.acquireDODS(NetcdfDataset.java:805)
```

In this case please contact the data supplier that created the NetCDF file and ask them to make the NetCDF file CF-1.6 compliant. If the file is not CF-1.6 compliant then FEWS cannot import the file and there is no reason to contact the FEWS support team.

Related modules and documentation

Internal

- Import Module
- Import Module configuration options
- Import data using OPeNDAP
- Available data types
- NETCDF-CF_GRID
- NETCDF-CF_TIMESERIES
- NETCDF-CF_PROFILE
- NETCDF-CF_TRAJECTORY

External

- http://cfconventions.org/
- https://github.com/ugrid-conventions/ugrid-conventions
- OPeNDAP
- THREDDS
- https://www.unidata.ucar.edu/software/netcdf/docs/BestPractices.html
- https://www.unidata.ucar.edu/software/netcdf/workshops/2012/cf/index.html
- https://www.unidata.ucar.edu/software/netcdf/workshops/2012/index.html
- http://zandmotor.citg.tudelft.nl/netcdfkickstarter/
- netCDF kickstarter