D-Flow FM adapter

Please note that for running DFlow-FM from Delft-FEWS only a pre-adapter is needed (a post-adapter is not needed).

D-Flow FM pre-adapter

Model pre-adapter for running D-Flow FM (D-Flow Flexible Mesh) model from Delft-FEWS.

For information about the D-Flow FM model see http://oss.deltares.nl/web/delft3d/d-flow-flexible-mesh

For the D-Flow FM user manual, D-Flow FM technical reference manual and other relevant manuals, see:
http://content.oss.deltares.nl/delft3d/manuals/D-Flow_FM_User_Manual.pdf
http://content.oss.deltares.nl/delft3d/manuals/D-Flow_FM_Technical_Reference.pdf
http://content.oss.deltares.nl/delft3d/manuals/

Usage: DFlowFMPreAdapter <netcdf run file pathname relative to current working directory>

Class name: nl.deltares.dflowfm.DFlowFMPreAdapter

Properties

model_id	(required)	Identifier of the model. This should be the first part of the .mdu file name. This is used to find the relevant .mdu file(s) and restart .nc file(s), as described below.
mdu_file	Deprecated. Do not use.	Pathname of the mdu file to update. This should be either an absolute path or a path relative to the workDir specified in the netcdf run file.
input_grid_files_to_convert	(optional) Deprecated. The latest version of D-Flow FM can directly read grid NetCDF files that are exported from Delft-FEWS. This works for the parameters wind, pressure, rainfall, radiation, temperature and humidity. Therefore converting grid files is not needed anymore.	One or more pathnames of netcdf files with input grid data that should be converted. The pathnames should be separated by semi-colons (;). Each pathname should be either an absolute path or a path relative to the workDir specified in the netcdf run file.

Notes for users

- For all files that are written by this adapter, if the file to be written already exists, then it will be overwritten.
- This program assumes that the model always runs in time zone GMT.
- This program writes log messages to a log file called dflowfm_pre_adapter_log.txt in the workDir specified in the netcdf run file.
- The pre-adapter uses the information in the specified netcdf run file as input for its activities (see below).

pre-adapter activities: Update MDU file(s)

In the found mdu file(s) the following entries will be updated automatically (no tags needed):

TStart	Start time of the model run, in units of Tunit relative to RefDate. Uses the Tunit and RefDate that are specified in the mdu file.
TStop	End time of the model run, in units of Tunit relative to RefDate. Uses the Tunit and RefDate that are specified in the mdu file.
RestartFile	Either * pathname of the input state file relative to the MDU file if the input state file is not empty (warm state start) or * empty string if the input state file is an empty dummy file of 0 bytes length (cold state start) or * empty string if there is no input state file at all (cold state start).
RestartDateTime	The restart time, this is set equal to the start time of the model run. Note: It seems that in D-Flow FM this RestartDateTime is overruled by the timestamp in the filename of the restart file. If this causes problems, then make sure that the filename of the restart file does not contain a timestamp.
RstInterval	Interval (in seconds) for writing *_rst.nc restart files. RstInterval is set equal to the model run duration, so that only one output restart file is written, exactly at the end of the model run. RstInterval can be overwritten by a property in the run info file called "restartIntervalForFm"

MDU file name format

The mdu file(s) to update should be in the workDir specified in the netcdf run file. This program only supports:

- a single mdu file with file name format *<model_id>.mdu* or
  Valid example: *gtsm.mdu*
- (in case of domain decomposition) one mdu file for each partition with file name format *<model_id>_<partition_number>.mdu*
  Valid example: *gtsm_0000.mdu, gtsm_0001.mdu, gtsm_0002.mdu*

Restart file name format

In case of a warm state start, there should be exactly one input state file (restart file) for each mdu file. This program only supports a single input state file with file name format

- *<model_id>_rst.nc* , or
  Valid example: *gtsm_rst.nc*
- *<model_id>_<timestamp>_rst.nc*
  Valid example: *gtsm_20160407_140000_rst.nc*

In case of domain decomposition there should be one input state file for each partition with file name format

- *<model_id>_<partition_number>_rst.nc, or*
  Valid example: *gtsm_0001_rst.nc*
- *<model_id>_<partition_number>_<timestamp>_rst.nc*
  Valid example: *gtsm_0001_20160407_140000_rst.nc*

pre-adapter activities: Unpdate external forcing files (optional)

The pre-adapter (version Delft-FEWS 2017.01.01 and upwards) converts exported scalar timeseries from xml to external forcing files (i.e. .bc and .tim files). Please note the following:

- The pre-adapter reads the .ext files mentioned in the .mdu file, and finds here the files to check for keywords
- See Delft3D adapter with NetCDF run file#Templatefilesandkeywords on how to use template files and keywords.
- The pre-adapter can handle at most two different forcing files, so if you list more in the .mdu file they will be ignored.
- The pre-adapter assumes a timestep in minutes. Originally, the .tim files could only handle minutes. The newer .bc files can use any unit, but the pre-adapter still assumes minutes. The pre-adapter does -not- interpret the Tunit as described in the .mdu file (Implementing this would be an improvement of the pre-adapter.)
- Example config files can be found elsewhere on this wiki page

pre-adapter activities: Convert input grid time series (optional)

The netcdf file(s) specified in the property "input_grid_files_to_convert" will be converted to files in arcinfo/curvi format. Each netcdf file will be converted to a file with the same path and name as the netcdf file but with a different extension (.amu, .amv or .amp). If the property "input_grid_files_to_convert" is not specified, then this step does nothing.

Conversion from variable to file extension

Each netcdf file should contain only one variable with grid data. A netcdf file with multiple variables with grid data results in an error. The extension of the created file depends on the name of the variable in the netcdf file. For example the original file input/x_wind.nc is converted to input/x_wind.amu

netcdf variable name	extension
x_wind	.amu
y_wind	.amv
air_pressure	.amp
precipitation	not supported by DFlowFM (for rainfall DFlowFM can use the netcdf file directly)
any other name	not supported by DFlowFM

Auxiliary grid file

The format (meteo_on_equidistant_grid/meteo_on_curvilinear_grid) of each of the created files depends on whether there is an auxiliary grid file present for that file. To use an auxiliary grid file for a given netcdf file, it must have the same path and name as the netcdf file, but a different extension (.grd). If an auxiliary grid file is present, then the netcdf file will be converted to a curvi file of type meteo_on_curvilinear_grid that refers to the auxiliary grid file. Otherwise it will be converted to an arcinfo file of type meteo_on_equidistant_grid. For rectangular and curvilinear grids there must always be an auxiliary grid file present, otherwise an error is given. For regular grids no auxiliary grid file is needed.

grid type	auxiliary grid (.grd) file needed	type of created file
regular	no	arcinfo file of type meteo_on_equidistant_grid
rectangular	yes	curvi file of type meteo_on_curvilinear_grid
curvilinear	yes	curvi file of type meteo_on_curvilinear_grid

Order of grid cells written

The order of the grid cell values in the arcinfo/curvi grid file created by this adapter depend on the type of grid (regular/rectangular/curvilinear). For regular grids (arcinfo meteo_on_equidistant_grid file format or curvi meteo_on_curvilinear_grid file format) the grid cells are always ordered per row from left to right, starting with the upper row of the grid. For rectangular grids (curvi meteo_on_curvilinear_grid file format) the grid cells are always ordered per row from left to right, starting with the upper row of the grid. For curvilinear grids (curvi meteo_on_curvilinear_grid file format) the grid cells are always in the same order as in the netcdf file, which depends in turn on the order of the grid cells in the corresponding grid definition in Delft-FEWS (if the file was exported from Delft-FEWS). If an auxiliary grid (.grd) file is used, then the grid cell coordinates in the .grd file must be in the same order as the grid cell values in the corresponding curvi file. The easiest way to accomplish this is to run the adapter once, then check the order of the grid cell values in the created curvi file(s), then manually make sure that the grid cell coordinates in the corresponding .grd file(s) are in the right order.

Coordinate system used

The coordinate system for the coordinates in an arcinfo meteo_on_equidistant_grid file created by this adapter depends on the coordinate system used in the netcdf file, which depends in turn on the coordinate system (geodatum) in the corresponding grid definition in Delft-FEWS (if the file was exported from Delft-FEWS). Need to manually make sure that this is the same coordinate system as the coordinate system used by the model.

D-Flow FM data in Delft-FEWS

3D data: sigma layer vs Z layer

At the moment there is no example yet of a z layer D-Flow FM model connected to Delft-FEWS. All documentation and examples below are related to sigma layer models. Z layers are supported by the model adapter though. An example of import and display of z layers in a Delft-3D model can be found at Delft3D adapter - 4.3 Import and display 3D data (z layers)

Export of 3D data in generalAdapter (z layers) - NETCDF-CF_ZLAYERS

Scalar time series at the same geo point Z but with different X,Y are considered to be a Z-layer. All available Z's are used to create a Z-axis (layer axis) in the NetCdf file, and the time series values are written to the associated Z element. To export scalar time series as Z_layers in GA with NETCDF-CF_ZLAYERS, use option **<exportZLayers>true</exportZLayers>** in the exportNetcdfActivity (see config example below).

An example for float salinity(time=5, node=26, z=40);

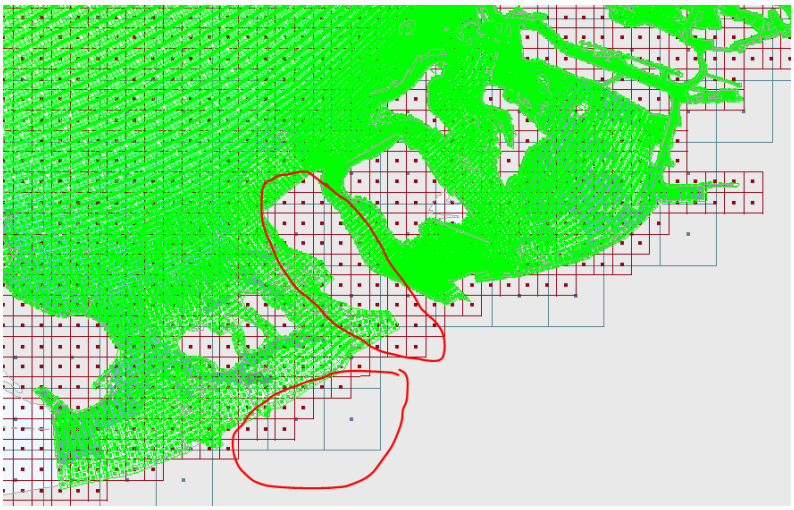
Values of Z-axis are stored in meters. Per parameter only one Z-axis is allowed. Different parameters may have different Z-axis. Z-axis values are sorted in ascending order. The number of stations in the nc file equals to the number of unique X,Y that are available in the scalar time series. The location id's /names associated with the first (lowest) Z are written to the nc file as **station id**'s/names. If there are parentLocations configured, then the IdMap can be used to write the **parentLocations id**'s to the nc file. By default the **long_name** attribute of the parameters is equal to the parameter id. This default behavior is overwritten with the configuration of a **parameter description** in Parameters.xml, which will be used as long_name in the nc file instead. In GA the default missing value for time series is -999. You can overwrite it in GA using **<missVal>**, for example <missVal>NaN</missVal>

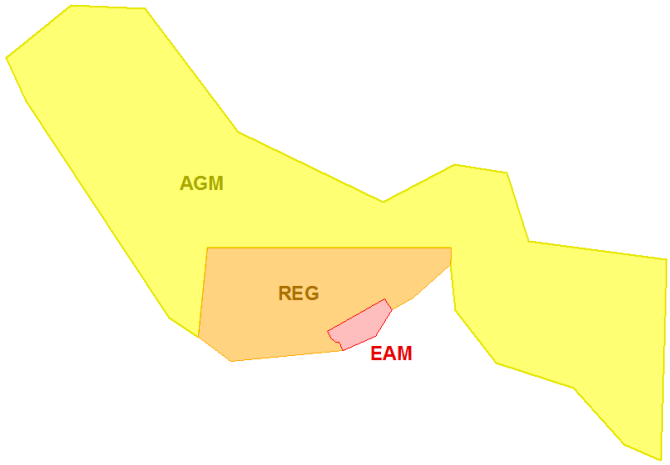
Import of 3D data in generalAdapater (sigma layers)

In case of sigma layer data, you need to configure a locationSet linking the sigma layers (% of depth) to a layer index, see config example below. In a way, this makes the handling of sigma and z layers in Delft-FEWS similar, since both can be referred to through a layer index.

Display of 2D data for overlapping domains

A common model set-up in D-Flow FM will make use of several models for an project area. For example, you could have a large scale coarse regional model, with 1 (or more) finer model(s) zooming in on the area of interest which is (are) located within the regional model area, i.e. the model areas overlap. When you display all models in 1 gridPlot, you don't want to see the coarse grid peaking out from underneath the finer local model(s). The example in the figure has 3 different model areas, going from coarse (blue grid), through intermediate (red grid) to a fine resolution (green grid).





In order to hide the coarser grids where a finer grid is available, you'll need to define a shapefile which masks the area of each model domain that you would like visualised in the gridDisplay. These shapefiles should (barely) overlap, see example in the figure on the right. All data is still available in the gridPlot, but only the data within the shapefile contour is displayed in the gridDisplay. When the user double-clicks in the gridDisplay, a timeSeriesDisplay with scalar data for all models with data for that location (in this example up to 3) is displayed, even though the data might be masked in the gridDisplay. Co nfig example below.

Display of 3D data (sigma layers)

In GridDisplay.xml you need to configure an additional sigmaScaleReferenceTimeSeriesSet for sigma layers (see config example below). A on the fly transformation allows the user to dynamically interpolate between sigma layers in the grid display. For more information on the GridDisplay visit 01 Grid Display - sigmaScaleReferenceTimeSeriesSet

System requirements

- This program needs Java version 8 or higher.
- This program needs the following Java libraries:
    - commons-httpclient-3.0.1.jar
    - Delft_Util.jar (revision 60330)
    - grib-8.0.jar
    - log4j-1.2.14.jar
    - netcdf-4.2.jar
    - slf4j-api-1.5.6.jar
    - slf4j-log4j12-1.5.6.jar
    - TimeSeriesImport.jar (revision 60330)

Configuration example

Please note that for running DFlow-FM from Delft-FEWS only a pre-adapter is needed (a post-adapter is not needed).

Update external forcing files

Example of changes to the config when dealing with a D-Flow FM model with scalar boundary conditions.

Excerpt from .mdu file related to external forcing

RegionConfigFiles/Grids.xml

```
[external forcing]
ExtForceFileNew   = mackay_bnd.ext      # ExtForceFileNew DDB uses new format
ExtForceFile      = mackay_pioneer.ext  # *.ext
```

Example .ext file with reference to .bc file:

RegionConfigFiles/Grids.xml

```
[boundary]
quantity       = waterlevelbnd
locationfile   = mackay_bnd.pli
forcingfile    = mackay_bnd.bc
```

With corresponing .pli file:

RegionConfigFiles/Grids.xml

```
mackay_bnd
1 2
1.49201410e+02 -2.10467553e+01 mackay_bnd_0001
```

and corresponding .bc file:

RegionConfigFiles/Grids.xml

```
[forcing]
Name                           = mackay_bnd_0001
Function                       = timeseries
Time-interpolation             = linear
Quantity                       = time
Unit                           = minutes since 2017-01-01 00:00:00
Quantity                       = waterlevelbnd
Unit                           = m
$(FLOW_TIMESERIES: waterlevelbnd/mackay_bnd_0001)
```

Example .ext file with .tim file:

RegionConfigFiles/Grids.xml

```
QUANTITY=discharge_salinity_temperature_sorsin
FILENAME=mackay_pioneer.pli
FILETYPE=9
METHOD=1
OPERAND=O
AREA=1
```

With corresponding .pli file:

RegionConfigFiles/Grids.xml

```
pioneer
1 2
1.49099320e+02 -2.11482830e+01
```

and corresponding .tim file:

RegionConfigFiles/Grids.xml

```
$(FLOW_TIMESERIES: Q_sim_fcst/pioneer)
```

Example of exportTimeSeriesActivity in General Adapter config file:

RegionConfigFiles/Grids.xml

```
<exportTimeSeriesActivity>
    <exportFile>timeseries.xml</exportFile>
    <timeSeriesSets>
        <timeSeriesSet>
            <moduleInstanceSetId>URBS_Forecast</moduleInstanceSetId>
            <valueType>scalar</valueType>
            <parameterId>Q.sim.fcst</parameterId>
            <locationSetId>DFLOWFM_river.$CATCHMENT$_$SUBCATCHMENT$</locationSetId>
            <timeSeriesType>simulated forecasting</timeSeriesType>
            <timeStep unit="minute" multiplier="15"/>
            <relativeViewPeriod unit="day" start="0" end="3"/>
            <readWriteMode>read only</readWriteMode>
        </timeSeriesSet>
        <timeSeriesSet>
            <moduleInstanceId>ImportROMS</moduleInstanceId>
            <valueType>scalar</valueType>
            <parameterId>H.tidal.fcst</parameterId>
            <locationSetId>DFLOWFM_coastal.$CATCHMENT$_$SUBCATCHMENT$</locationSetId>
            <timeSeriesType>external forecasting</timeSeriesType>
            <timeStep unit="minute" multiplier="30"/>
            <readWriteMode>read complete forecast</readWriteMode>
        </timeSeriesSet>
    </timeSeriesSets>
</exportTimeSeriesActivity>
```

D-Flow FM model configuration example (single domain, 2D data import)

Example of a FEWS general adapter configuration that uses the DFlow-FM adapter.

Single domain example config General Adapter

```
<?xml version="1.0" encoding="UTF-8"?>
<generalAdapterRun 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/generalAdapterRun.xsd">
        <general>
                <rootDir>$REGION_HOME$\Modules\dflowfm</rootDir>
                <workDir>%ROOT_DIR%</workDir>
                <exportDir>%ROOT_DIR%\dflowfm_curacao\input</exportDir>
                <exportDataSetDir>$REGION_HOME$\Modules</exportDataSetDir>
                <exportIdMap>IdExport_DFlowFM</exportIdMap>
                <importDir>%ROOT_DIR%\dflowfm_curacao\output</importDir>
                <importIdMap>IdImport_DFlowFM</importIdMap>
                <dumpFileDir>$GA_DUMPFILEDIR$</dumpFileDir>
                <dumpDir>%ROOT_DIR%\dflowfm_curacao\dump</dumpDir>
                <diagnosticFile>%ROOT_DIR%\dummy.xml</diagnosticFile>
                <missVal>-999.</missVal>
```

```
                <!-- Take care this should be the timezone the computer is running in -->
                <timeZone>
                        <timeZoneOffset>-04:00</timeZoneOffset>
                </timeZone>
                <endDateTimeFormat>yyyyMMdd_HHmmss</endDateTimeFormat>
        </general>
        <activities>
                <startUpActivities>
                        <purgeActivity>
                                <filter>%ROOT_DIR%\dflowfm_curacao\output\*.*</filter>
                        </purgeActivity>
                        <purgeActivity>
                                <filter>%ROOT_DIR%\dflowfm_curacao\input\*.*</filter>
                        </purgeActivity>
                </startUpActivities>
                <exportActivities>
                        <exportStateActivity>
                                <moduleInstanceId>DFlowFM_curacao_Historical</moduleInstanceId>
                                <stateExportDir>%ROOT_DIR%\dflowfm_curacao\instate</stateExportDir>
                                <stateSelection>
                                        <warmState>
                                                <stateSearchPeriod unit="hour" start="-23" end="-2"/>
                                        </warmState>
                                </stateSelection>
                        </exportStateActivity>
                        <exportNetcdfActivity>
                                <exportFile>air_pressure.nc</exportFile>
                                <timeSeriesSets>
                                        <timeSeriesSet>
                                                <moduleInstanceId>WFLOW_curacao_GA_Historical</moduleInstanceId>
                                                <valueType>grid</valueType>
                                                <parameterId>P.specific</parameterId>
                                                <locationId>wflow_curacao</locationId>
                                                <timeSeriesType>simulated historical</timeSeriesType>
                                                <timeStep unit="hour" multiplier="1"/>
                                                <relativeViewPeriod unit="hour" end="0"/>
                                                <readWriteMode>add originals</readWriteMode>
                                        </timeSeriesSet>
                                </timeSeriesSets>
                        </exportNetcdfActivity>
                        <exportNetcdfRunFileActivity>
                                <description>This run file is passed as argument to DFLOWFM pre adapter<
/description>
                                <exportFile>%WORK_DIR%\run_info.nc</exportFile>
                                <properties>
                                        <string key="model_id" value="dflowfm_curacao\curacao"/>
                                        <string key="input_grid_files_to_convert" value="%ROOT_DIR%
\dflowfm_curacao\input\x_wind.nc;%ROOT_DIR%\dflowfm_curacao\input\air_pressure.nc"/>
                                </properties>
                        </exportNetcdfRunFileActivity>
                </exportActivities>
                <executeActivities>
                        <executeActivity>
                                <description>DFlowFM pre adapter</description>
                                <command>
                                        <className>nl.deltares.dflowfm.DFlowFMPreAdapter</className>
                                        <binDir>adapter</binDir>
                                </command>
                                <arguments>
                                        <argument>%WORK_DIR%\run_info.nc</argument>
                                </arguments>
                                <logFile>
                                        <file>%WORK_DIR%\dflowfm_pre_adapter_log.txt</file>
                                        <errorLinePattern>*ERROR*</errorLinePattern>
                                        <warningLinePattern>*WARN*</warningLinePattern>
                                        <infoLinePattern>*INFO*</infoLinePattern>
                                        <debugLinePattern>*DEBUG*</debugLinePattern>
                                </logFile>
                                <timeOut>99999999</timeOut>
                                <ignoreDiagnostics>true</ignoreDiagnostics>
                        </executeActivity>
```

```
                        <executeActivity>
                                <description>Run DFLOWFM</description>
                                <command>
                                        <executable>bin\unstruc.exe</executable>
                                </command>
                                <arguments>
                                        <argument>--autostartstop</argument>
                                        <argument>dflowfm_curacao\curacao.mdu</argument>
                                </arguments>
                                <logFile>
                                        <file>%WORK_DIR%\dflowfm_curacao\curacao.dia</file>
                                        <errorLinePattern>*ERROR*</errorLinePattern>
                                        <warningLinePattern>*WARNING*</warningLinePattern>
                                        <debugLinePattern>*INFO*</debugLinePattern>
                                        <debugLinePattern>*DEBUG*</debugLinePattern>
                                </logFile>
                                <timeOut>44200000</timeOut>
                                <ignoreDiagnostics>true</ignoreDiagnostics>
                        </executeActivity>
                </executeActivities>
                <importActivities>
                        <importStateActivity>
                                <stateFile>
                                        <importFile>%WORK_DIR%\dflowfm_curacao\output\curacao_%END_DATE_TIME%
_rst.nc</importFile>
                                        <relativeExportFile>curacao_%END_DATE_TIME%_rst.nc</relativeExportFile>
                                </stateFile>
                        </importStateActivity>
                        <importNetcdfActivity>
                                <importFile>%WORK_DIR%\dflowfm_curacao\output\curacao_map.nc</importFile>
                                <timeSeriesSets>
                                        <timeSeriesSet>
                                                <moduleInstanceId>DFlowFM_curacao_Historical</moduleInstanceId>
                                                <valueType>grid</valueType>
                                                <parameterId>H.sim</parameterId>
                                                <locationId>DFlowFM_Curacao</locationId>
                                                <timeSeriesType>simulated historical</timeSeriesType>
                                                <timeStep unit="nonequidistant"/>
                                                <readWriteMode>add originals</readWriteMode>
                                        </timeSeriesSet>
                                </timeSeriesSets>
                        </importNetcdfActivity>
                </importActivities>
        </activities>
</generalAdapterRun>
```

Z layers - export D-Flow FM 3D results (NETCDF-CF_ZLAYERS)

Sigma layer example config General Adapter import activity

```
    <activities>
        <exportActivities>
            <exportNetcdfActivity>
                <exportFile>hycom_boundary.nc</exportFile>
                <exportZLayers>true</exportZLayers>
                <timeSeriesSets>
                    <timeSeriesSet>
                        <moduleInstanceId>Interpolate_Boundaries_HYCOM_forecast</moduleInstanceId>
                        <valueType>scalar</valueType>
                        <parameterId>S.simulated</parameterId>
                        <locationSetId>HYCOM.Boundaries_AllLayers</locationSetId>
                        <timeSeriesType>simulated forecasting</timeSeriesType>
                        <timeStep unit="hour" multiplier="3"/>
                        <relativeViewPeriod unit="day" start="0" end="1" endOverrulable="true"/>
                        <readWriteMode>read only</readWriteMode>
                    </timeSeriesSet>
                </timeSeriesSets>
                <timeSeriesSets>
                    <timeSeriesSet>
                        <moduleInstanceId>Interpolate_Boundaries_HYCOM_forecast</moduleInstanceId>
                        <valueType>scalar</valueType>
                        <parameterId>T.simulated</parameterId>
                        <locationSetId>HYCOM.Boundaries_AllLayers</locationSetId>
                        <timeSeriesType>simulated forecasting</timeSeriesType>
                        <timeStep unit="hour" multiplier="3"/>
                        <relativeViewPeriod unit="day" start="0" end="1" endOverrulable="true"/>
                        <readWriteMode>read only</readWriteMode>
                    </timeSeriesSet>
                </timeSeriesSets>
            </exportNetcdfActivity>
        </exportActivities>
        </activities>
```

Sigma layers - import D-Flow FM 3D results

Example of changes to the config when dealing with a D-Flow FM **model with multiple domains**, using sigma layers.

RegionConfigFiles/Grids.xml

```
<irregular locationId="sigma_0000">
    <netcdfFile>
        <file>$TEMP_DIR$/3d_flowgeom_0000.nc</file>
        <meshTopologyVariableName>mesh2d</meshTopologyVariableName>
        <staggerLocation>face</staggerLocation>
    </netcdfFile>
</irregular>
```

RegionConfigFiles/Locations.xml

```
<location id="sigma.0000">
    <x>0</x>
    <y>0</y>
</location>
```

RegionConfigFiles/LocationSets.xml

```
<locationSet id="sigmaLayers_0000">                                                    <!-- i.e. locationSet for
all sigmalayers in domain 0000 -->
    <csvFile>
        <file>sigma_layers.csv</file>
        <id>SigmaLayer_%INDEX%_0000</id>                                        <!-- i.e. locationId consists
of SigmaLayer_0_0000, SigmaLayer_1_0000, etc for all indices / sigma layers in domain 0000 -->
        <parentLocationId>sigma.0000</parentLocationId>                <!-- parentLocationId, through which all
sigmaLayers / locations in this set can be resolved, see for example the GridDisplay (see example below) -->
        <x>0</x>
        <y>0</y>
        <layerSigmaCoordinate>%SIGMA%</layerSigmaCoordinate>
    </csvFile>
</locationSet>
```

MapLayerFiles/sigma_layers.csv

```
INDEX,SIGMA
0,0.025
1,0.075
2,0.125
3,0.175
4,0.225
5,0.275
6,0.325
7,0.375
8,0.425
9,0.475
10,0.525
11,0.575
12,0.625
13,0.675
14,0.725
15,0.775
16,0.825
17,0.875
18,0.925
19,0.975
```

Through the use of <mapLocationsByLayerSigmaCoordinate> the location is resolved by the sigma layer coordinates in the generalAdapter:

Sigma layer example config General Adapter import activity

```
<?xml version="1.0" encoding="UTF-8"?>
<generalAdapterRun 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/generalAdapterRun.xsd">
    <general>
        <rootDir>$TEMP_DIR$</rootDir>
        <workDir>$TEMP_DIR$</workDir>
        <exportDir>$TEMP_DIR$</exportDir>
        <importDir>$TEMP_DIR$</importDir>
        <dumpFileDir>$TEMP_DIR$</dumpFileDir>
        <dumpDir>$TEMP_DIR$</dumpDir>
        <diagnosticFile>$TEMP_DIR$/diagnostic.xml</diagnosticFile>
    </general>
        <activities>
                <importActivities>
                        <importNetcdfActivity>
                <importFile>%WORK_DIR%/DFM_OUTPUT/model_0000_map.nc</importFile>
                <timeSeriesSets>
                    <timeSeriesSet>
                        <moduleInstanceId>model_DFlowFM_FC</moduleInstanceId>
                        <valueType>grid</valueType>
                        <parameterId>H.simulated</parameterId>
                        <locationId>sigmaLayers.0000<
/locationId>                                                                    <!-- parentLocation for
domain 0000 -->
                        <timeSeriesType>simulated forecasting</timeSeriesType>
                        <timeStep unit="nonequidistant"/>
                        <readWriteMode>add originals</readWriteMode>
                        <expiryTime unit="day" multiplier="$EXPIRY_DAYS_SIMULATED_FORECAST_GRID$"/>
                    </timeSeriesSet>
                    <timeSeriesSet>
                        <moduleInstanceId>model_DFlowFM_FC</moduleInstanceId>
                        <valueType>grid</valueType>
                        <parameterId>T.simulated.v</parameterId>
                        <locationSetId>sigmaLayers_0000<
/locationSetId>                                                         <!-- locationSet off all
sigmaLayers linked to above used parentLocation -->
                        <timeSeriesType>simulated forecasting</timeSeriesType>
                        <timeStep unit="nonequidistant"/>
                        <readWriteMode>add originals</readWriteMode>
                        <expiryTime unit="day" multiplier="$EXPIRY_DAYS_SIMULATED_FORECAST_GRID$"/>
                    </timeSeriesSet>
                </timeSeriesSets>
                <mapLocationsByLayerSigmaCoordinate>true</mapLocationsByLayerSigmaCoordinate>
            </importNetcdfActivity>

<importNetcdfActivity>
<!-- repeat for output files of all domains -->
                            <importFile>%WORK_DIR%/DFM_OUTPUT/model_0001_map.nc</importFile>
                            <timeSeriesSets>
                                <timeSeriesSet>
                                    <moduleInstanceId>model_DFlowFM_FC</moduleInstanceId>
                                    <valueType>grid</valueType>
                                    <parameterId>H.simulated</parameterId>
                                    <locationId>sigmaLayers.0001</locationId>
                                    <timeSeriesType>simulated forecasting</timeSeriesType>
                                    <timeStep unit="nonequidistant"/>
                                    <readWriteMode>add originals</readWriteMode>
                                    <expiryTime unit="day" multiplier="$EXPIRY_DAYS_SIMULATED_FORECAST_GRID$"/>
                                </timeSeriesSet>
                                        ...
                            </timeSeriesSets>
                            <mapLocationsByLayerSigmaCoordinate>true</mapLocationsByLayerSigmaCoordinate>
                        </importNetcdfActivity>
            </importActivities>
        </activities>
</generalAdapterRun>
```

Sigma layers - display D-Flow FM 3D results in GridDisplay

This example builds on the grid, location and locationSet defined in the example above. It assumes sigma layers for a **multi domain model**.

RegionConfigFiles/LocationSets.xml

```
<locationSet id="SigmaLayer_0_####">                                <!-- SigmaLayer with index 0 for all
domains -->
    <locationId>SigmaLayer_0_0000</locationId>
    <locationId>SigmaLayer_0_0001</locationId>
    <locationId>SigmaLayer_0_0002</locationId>
        ...
</locationSet>
<locationSet id="sigma.merged">                                        <!-- parentLocations for all domains -->
    <locationId>sigma.0000</locationId>                        <!-- parentLocationId for domain 0000 -->
    <locationId>sigma.0001</locationId>
    <locationId>sigma.0002</locationId>
        ...
</locationSet>
```

Sigma layer example config GridDisplay

```
<gridDisplay 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/gridDisplay.xsd">
<title>title</title>
<gridPlotGroup id="gridPlotGroupId" name="gridPlotGroupName">
        <gridPlot id="gridPlotId" name="gridPlotName">
                <dataLayer>
                        <uTimeSeriesSet>
                                <moduleInstanceId>DFlowFM_HC</moduleInstanceId>
                                <valueType>grid</valueType>
                                <parameterId>parameter</parameterId>
                                <locationSetId>SigmaLayer_0_####</locationSetId>                <!-- Configure
only the top layer (index = 0). The sibling locations (i.e. other sigma layers) are resolved through the parent
of the top layer. -->
                                <timeSeriesType>simulated historical</timeSeriesType>
                                <timeStep unit="nonequidistant"/>
                                <readWriteMode>read complete forecast</readWriteMode>
                        </uTimeSeriesSet>
                        <vTimeSeriesSet>
                                <moduleInstanceId>DFlowFM_HC</moduleInstanceId>
                                <valueType>grid</valueType>
                                <parameterId>parameter</parameterId>
                                <locationSetId>SigmaLayer_0_####</locationSetId>                <!-- Configure
only the top layer (index = 0). The sibling locations (i.e. other sigma layers) are resolved through the parent
of the top layer. -->
                                <timeSeriesType>simulated historical</timeSeriesType>
                                <timeStep unit="nonequidistant"/>
                                <readWriteMode>read complete forecast</readWriteMode>
                        </vTimeSeriesSet>

<sigmaScaleReferenceTimeSeriesSet>                                                          <!-- when this is
configured, a vertical slider becomes automatically visible in GridDisplay to slide through the water column -->
                                <moduleInstanceId>DFlowFM_HC</moduleInstanceId>
                                <valueType>grid</valueType>
                                <parameterId>parameter</parameterId>
                                <locationSetId>sigma.merged</locationSetId>                         <!--
parentLocations for all domains, linking to all sigma layers -->
                                <timeSeriesType>simulated historical</timeSeriesType>
                                <timeStep unit="nonequidistant"/>
                                <readWriteMode>read complete forecast</readWriteMode>
                        </sigmaScaleReferenceTimeSeriesSet>
                </dataLayer>
                <verticalSliderRange start="0" end="100"/>                                                  <!--
limit the min and max water depth used in the vertical slider in the GridDisplay -->

<!-- if not configured the range of slider is automatically set to cover all available water depths in the grid
for the entire period displayed -->
  </gridPlot>
</gridPlotGroup>
```

Masking - display D-Flow FM 3D results in GridDisplay (multiple overlapping models)

When you have multiple overlapping models and you want to control which data is displayed in the gridDisplay, you can make use of a masking shapefile. Link this file to a locationSet. The <id> specified in the <esriShapeFile> config can be used in the GridDisplay, which will mask the data shown. For more information see above Display of 2D data for overlapping domains

RegionConfigFiles/LocationSets.xml

```
<locationSet id="wave_clipper.shp">
    <esriShapeFile>
        <file>wave_clipper.shp</file>
        <id>Wave</id>
        <x>0</x>
        <y>0</y>
    </esriShapeFile>
</locationSet>
```

DisplayConfigFiles/GridDisplay.xml

```
                    <valueTimeSeriesSet>
                        <moduleInstanceId>Wave_HC</moduleInstanceId>
                        <valueType>grid</valueType>
                        <parameterId>Wave.height.simulated</parameterId>
                        <locationId>Wave</locationId>
                        <timeSeriesType>simulated historical</timeSeriesType>
                        <timeStep multiplier="1" unit="hour"/>
                        <readWriteMode>read complete forecast</readWriteMode>
                    </valueTimeSeriesSet>
```