22-2 Export to Deltares Open Archive

The Delft-FEWS ExportArchiveModule

Within Delft-FEWS, the datasets are exported to the archive in a workflow using the ExportArchiveModule. This workflow needs to be scheduled on a regular interval to archive all relevant data. Simulated data sets are preferably exported to the archive from the workflow that created the simulation. This is especially the case when modifiers are used in the simulation. When a simulation is not exported to the archive directly, it might be that some modifiers that were used in the simulation are deleted or changed in the mean time. This can then result in archived simulation data without the modifiers that were used to produce them.

To prevent dependencies on other processes, the ExportArchiveModule is envisioned to write directly into the archive file storage. The FSS thus needs to be able to have write access to those disks.

For each kind of dataset, the ExportArchiveModule checks the database for changes over a (configured) relative period. It exports all data that meets the export instructions and that has changed within this period. Datasets are archived in a predefined directory structure, which is based on areaId, date and dataset name.

The schema of the associated configuration file (Figure 4.1) is defined at:
http://fews.wldelft.nl/schemas/version1.0/exportArchiveModule.xsd.

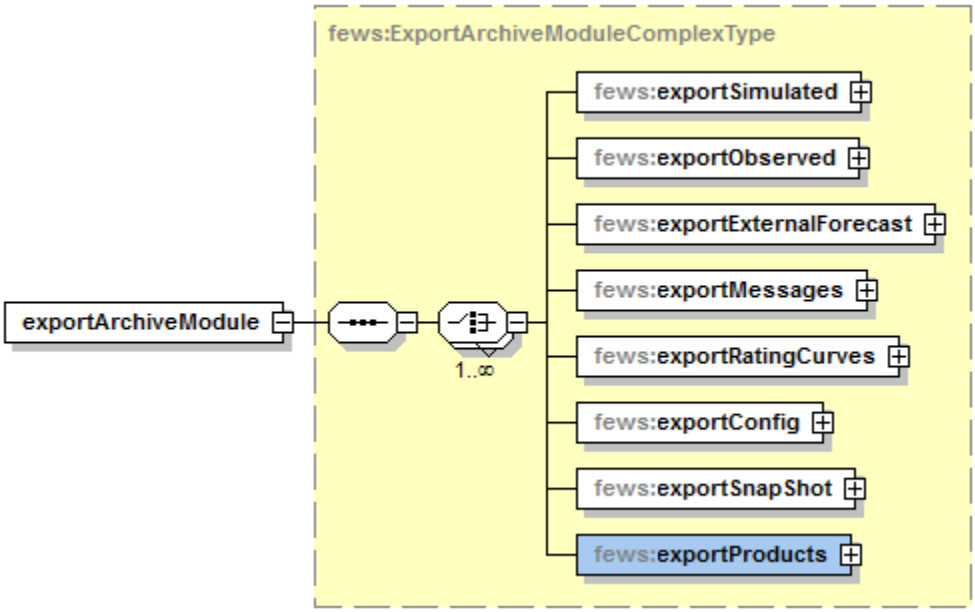


Figure 4.1 Top level of Delft-FEWS exportArchiveModule.xsd

- The Delft-FEWS ExportArchiveModule
- Archiving to File Storage
  - Archiving by using time series sets or workflow selection
  - Observations archiving by Delft-FEWS
  - Amalgamate observations
  - Messages archiving by Delft-FEWS
  - External forecast archiving by Delft-FEWS
  - Simulations archiving by Delft-FEWS
  - Configuration archiving by Delft-FEWS
  - Archiving rating curves by Delft-FEWS
  - Archiving Delft-FEWS snap shots
  - Archiving Delft-FEWS products
- Archiving to Archive Database
  - Archiving observed data to the archive database
  - Archiving simulated data to the archive database
  - Archiving external forecast data to the archive database

Archiving to File Storage

Archiving by using time series sets or workflow selection

The FEWS archive exporting module has several options for exporting data. In all those exports, time series can be selected by either specifying the time series using time series sets or by specifying a workflow in combination with a time series filter. When specifying a workflow, all the time series that comply with the defined filter will be exported. This filter can be used to select the desired data type, parameter, location, qualifier, etc. for exporting. By default the filter will be applied to the time series of the current workflow. If a workflowId is defined then the data from the last or current run will be used. One can also define a period, whereby all simulations or external forecasts for the that period are be exported.

Generated file name

The paths of the folders to which the data will be written is the same for both options, following the default folder structure for the respective data type.

When exports are setup by specifying timeseries, one must configure the filename of the exported file. If the workflow selection option is used to select which time series will be archived, it is not necessary to define the file names of the netcdf files in which the data will be archived. These names will be generated based on the timeseries that are stored in the file. The generated filename will be determined as follows:

<parameterid>_<qualifierIds>_<timestep>_<valuetype>.nc

A file containing scalar data for parameter Hobs with qualifier Q and time step SETS60 will be stored in file Hobs_Q_SETS60_scalar.nc. It is also possible to configure a fixed prefix for each generated file name.

If timeseries only differ by module instance id then the filename is extended with the module instance id.

Example

Below an example of an export for a workflow with time series selection through a filter. In the workflowTimeSeriesSelection section, one must specify at least one filter and one workflowId (optional).

```
<exportArchiveModule xsi:schemaLocation="http://www.wldelft.nl/fews http://fews.wldelft.nl/schemas/version1.0
/exportArchiveModule.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.wldelft.nl
/fews">
        <exportExternalForecast>
                <general>
                        <archiveFolder>$ARCHIVE_DIR$</archiveFolder>
                        <idMapId>IdExportArchiveExternalForecasts</idMapId>
                </general>
                <activities>
                        <netcdfExportActivities>
                                <netcdfExportActivity>
                                        <fileNamePrefix>Deltares</fileNamePrefix>
                                        <areaId>test</areaId>
                                        <sourceId>test</sourceId>
                                        <includeComments>false</includeComments>
                                        <includeFlags>false</includeFlags>
                                        <useModuleInstanceIdAsSourceId>true</useModuleInstanceIdAsSourceId>
                                        <workflowTimeSeriesSelection>
                                                <timeSeriesFilter>
                                                        <parameterId>Q</parameterId>
                                                </timeSeriesFilter>
                                            <workflowId>some_sub_workflow</workflowId>
                                        </workflowTimeSeriesSelection>
                                </netcdfExportActivity>
                        </netcdfExportActivities>
                </activities>
        </exportExternalForecast>
</exportArchiveModule>
```

Observations archiving by Delft-FEWS

For observations a dataset is generated for every area on a daily basis. The associated directory structure of the Delft-FEWS export for this type of data set is as follows:
<archive root>/<yyyy>/<MM>/<areaId>/<dd>/observed/
When Delft-FEWS generates the netcdf file, data is written to the same data block when the entire matrix is filled, i.e. all time steps are regular and none of the values is missing. For those locations with irregular time stamps, or missing values, a separate data block is used.
Within the netCDF file, each data block is accompanied by a header. Within each header a metadata item called 'timeseries_sets_xml' is included, holding the exact definition of the timeSeriesSet as the data was stored in the FEWS database. This feature allows full reproducibility of the time series via an Import workflow in a Delft-FEWS stand alone application, assuming that the associated Delft-FEWS configuration is in place.

The relative period used in the export of observed data sets requires some additional explanation. The relative period defines which daily data sets are written to archive. The export of observed data is usually scheduled once a day after midnight when all the data for the prior day are imported and available in FEWS. Because it is common to configure a data export of several days this means that data sets which are written earlier will be overwritten. This is done for several reasons. First all this ensure that when a data export was not executed for several days because of an system error this missing data sets will be added during the next run. In addition this will also allow that data edits will be stored in the archive. To ensure that data edits will be stored in the archive you must be sure that time window which you configure in your relative period is large enough to capture the data edits. However it very important that the relative period is not too large! If the beginning of the relative period starts at a time that data is already expired from the archive you will write missing values to the archive and erase the data in your archive! A relative period of -10 to -1 is usually sufficient for most systems.

The exportArchiveModule.xsd has a dedicated exportObserved section to configure the observed timeseries that need to be archived (see Figure 4.2). Table 4.1 documents the associated elements:

Table 4.1 Deft-FEWS export configuration for archiving observations

Element	Format	Description
general ComplexType		
dataSetPeriod	string	By default the daily data sets are created. It also possible to create monthly data sets. For operational purposes it is recommend to use daily data sets. For migrating data from other systems to the FEWS archive it might be convenient to export the data in montly data sets.
archiveFolder	string	Export destination folder, assumes that the account running the FEWS (FSS) application has write access
relativePeriod		Exports entire the dataset by day.
idMap	string	idMap applied to translate internal FEWS identifiers to identifiers that meet NetCDF-CF criteria.E.g. netcdf does not allow a full stop ('.') in the variable name
unitConversionsId	string	Id of UnitConversions to be used for unit mapping. Available since 2022.02, optional.
omitMissingValues	bool	Available since 2022.02. If set to true, missing values won't be written to the export file. Optional, default is false.
ignoreNonExistingLocationSets	**bool**	**If this option is set FEWS will not log an error when an location set is not configured.**
verifyExportedTimeSeries	bool	if this option is set FEWS will verify if data in the exported netcdf is the same as in the FEWS database.
netcdfObservedExportActivities ComplexType		
fileName	string	should include nc extension, otherwise files will not be read. preferably no spaces
areaId	string	area to which the dataset belongs
sourceId	string	defines the source of the data set
exportLocationAttributeAsNetCDFVariable	complex type	Since 2021.01, for archiving observed data. Adds a variable to the NetCdf file. Name of the variable is the value of ncVariable, the value is the configured location attribute of attributeId.
ncMetaData	string elem.	optional metadata tags within NetCDF file following CF convention. Supported by the internal catalogue of the THREDDS Data Server
includeFlags	bool	default=TRUE; if TRUE, a list of flags is stored, each value pointing to the associated flag
includeTimeSeriesProperties	bool	default=TRUE, if TRUE the properties of the time series will also be stored in the NetCDF file.
includeComments	bool	default=TRUE; if TRUE, a list of comments is stored, each value pointing to the associated comment
thresholdGroupId	string	identifies FEWS ThresholdGroup which is used to detect threshold crossings to be highlighted in the metaData.xml
timeSeriesSets		FEWS timeseries sets
workflowTimeSeriesSelection		In this tag the time series can selected by using a time series filter from either the current workflow or 1 or more other workflows.

*When an existing file is locked while it needs to be overwritten, the export function writes a new temporary file. The FileSweeper, a scheduled process, renames this file when the lock is removed from the original file.

Note: Not all information from observed (external historical) time series is archived. The data flags can be archived by setting the element "includeFlags" to true. The flag sources and users that made edits to the data is currently not archived.

Amalgamate observations

With daily exports of observed data per area, the number of small datasets on the file system may increases quickly. This has several disadvantages as the harvesting process will take longer processing many small datasets, while the performance of seamless integration (and webservices) will also drop. Therefore, an amalgamate process is developed which merges the daily observation files into one observation file per month. Currently, this amalgamate process is executed from the Fews-application side by an FSS by using the following configuration. Specify the relativePeriod in such way that the observed datafiles are stable and are not updated anymore. E.g. use one or two months of delay. There is no use in specifying a *startOverrulable:* when manually starrting this workflow it will only preocess the configured *relativePeriod.*

example amalgamate observations

```
<exportArchiveModule 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/exportArchiveModule.
xsd">
<amalgamateObserved>
    <general>
            <archiveFolder>$ARCHIVE_FOLDER$</archiveFolder>
        <relativePeriod unit="day" start="-62" end="-31"></relativePeriod>
    </general>
    <activities>
        <amalgamateObservedData>
            <areaId>$BASIN$</areaId>
        </amalgamateObservedData>
    </activities>
</amalgamateObserved>
</exportArchiveModule>
```

Element	Format	Description
GeneralAmalgamateObservedSection ComplexType		
archiveFolder	string	Export destination folder, assumes that the account running the FEWS (FSS) application has write access
relativePeriod		period experiencing changed data that should be amalgamted
AmalgamateActivities ComplexType		
amalgamateObservedData	AmalgamateActivity	indicator that observed data will be amalgamated
areaId	string	area for which amalgamation takes place

In future, the amalgamate process will be executed on the archive serverside via the ArchiveAdminConsole.

Messages archiving by Delft-FEWS

Delft-FEWS can export messages to the archive via the ArchiveExportModule (exportMessages activity).
For messages a dataset is generated for every area on a daily basis. The associated directory structure of the Delft-FEWS export for this type of dataset is as follows.
<archiveRoot>/<yyyy>/<MM>/<areaId>/<dd>/messages/

Note: The date associated with a message is the **creation time**, not the T0 at which the forecaster note was created or the Event Time. This means that the relative period as configured in exportMessages (which is relative to T0), for exporting the forecaster Notes to the Archive should cover the creation time. This is not so mch an issue in a live system, as it is in a Stand Alone, where T0 and System Time might be far apart.

The exportArchiveModule.xsd has a dedicated exportMessages section to configure the messages that need to be archived (see Figure 4.3). Currently only ForecasterNotes can be archived. Table 4.2 documents the associated elements:
Table 4.2 Delft-FEWS export configuration for archiving messages

Element	Format	Description
general ComplexType		
archiveFolder	string	Export destination folder, assumes that the account running the FEWS (FSS) application has write access
relativePeriod		Exports entire dataset by day, for any day where a database change (blob creation time) is detected within the relativePeriod (relative to T0). Existing files are overwritten*
MessagesActivities Complex Type		
forecasterNotesExportActivity ComplexType*		
areaId	string	area for which messages need to be archived

*When an existing file is locked while it needs to be overwritten, the export function writes a new temporary file. The FileSweeper, a scheduled process, renames this file when the lock is removed from the original file.

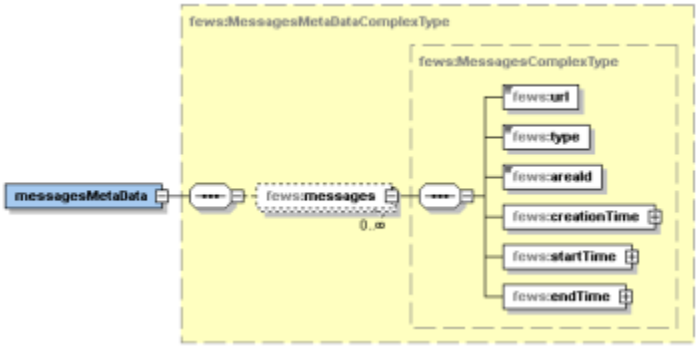


Figure 4.3 Delft-FEWS export configuration for archiving messages

External forecast archiving by Delft-FEWS

Delft-FEWS can export external forecast time series to the archive via the ArchiveExportModule (exportExternalForecast activity).
For external forecasts a dataset is generated for every area for every source for every forecast. The associated directory structure of the Delft-FEWS export for this type of dataset is as follows:
<archiveRoot>/<yyyy>/<MM>/<areaId>/<dd> /external_forecasts/<sourceId>_<ExtForecastTime>
The exportArchiveModule.xsd has a dedicated exportExternalForecast section to configure the time series that need to be archived (see Figure 4.4). Table 4.3 documents the associated elements:

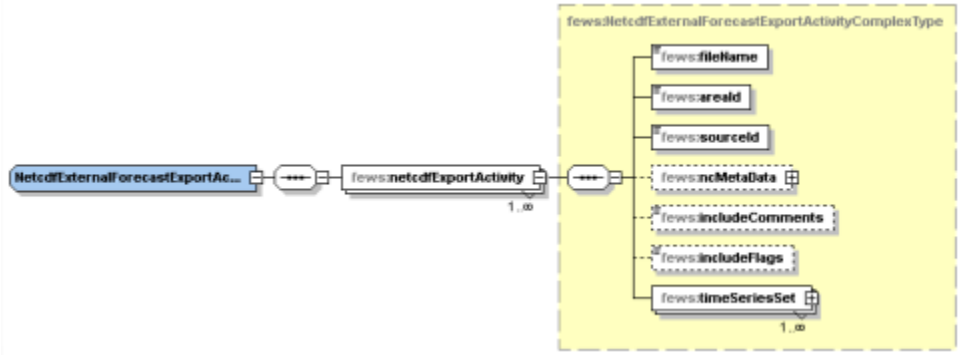


Figure 4.4 Delft-FEWS export configuration for archiving external forecasts

Table 4.3 Delft-FEWS export configuration for archiving external forecasts

Element	Format	Description

GeneralExportForecastSection ComplexType		
archiveFolder	string	Export destination folder, assumes that the account running the FEWS (FSS) application has write access
relativePeriod		Exports entire external forecast dataset by source and forecast time, for any forecast where a database change (blob creation time) is detected within the relativePeriod (relative to T0). Existing timeseries files are overwritten* If no relativePeriod is specified the external forecast with the latest forecast time is exported
idMap	string	idMap applied to translate internal FEWS identifiers to identifiers that meet NetCDF-CF criteria.E.g. netcdf does not allow a full stop ('.') in the variable name
ExternalForecastActivities ComplexType		
NetcdfExternalForecast ExportActivities ComplexType		
NetcdfExternalForecast ExportActivity ComplexType*		
fileName	string	without nc extension, preferably no spaces
areaId	string	area to which the dataset belongs
sourceId		identifies underlying source e.g. NWP product
ncMetaData	string elem.	optional metadata tags within NetCDF file following CF convention. Supported by the internal catalogue of the THREDDS Data Server
includeFlags	bool	only applied for scalar values. default=TRUE; if TRUE, a list of flags is stored, with each value pointing to the associated flag
includeComments	bool	only applied for scalar values. default=TRUE; if TRUE, a list of comments is stored, with each value pointing to the associated comment
includeTimeSeriesProperties	bool	only applied for scalar values. default=TRUE; if TRUE, a list of properties is stored, with each value pointing to the associated property
includeTimeRanges	bool	only applied for nonequidistant scalar values. Not valid for ensembles. default = TRUE; if TRUE, time-start and time-end properties will be stored if available.
includeValueRanges	bool	only applied for nonequidistant scalar values. Not valid for ensembles. default = TRUE; if TRUE, value-start and value-end properties will be stored if available.
timeSeriesSets		FEWS timeseries sets

*When an existing file is locked while it needs to be overwritten, the export function writes a new temporary file. The FileSweeper, a scheduled process, renames this file when the lock is removed from the original file.

Simulations archiving by Delft-FEWS

Delft-FEWS can export simulations to the archive via the ArchiveExportModule (exportSimulations activity).

Exporting forecasts can be done in three ways:

Exporting forecasts in the workflow which computed the forecast

The preferred way of archiving forecasts is at the end of the workflow which computed the forecast. Especially when you have using modifiers in your workflow. If you export the data in another workflow after the forecast is computed it is not guaranteed that the applied modifiers are still available in FEWS.

Exporting all forecasts in a relative period

It is also possible to archive a simulation from a scheduled workflow. In this case a relative period (relativePeriod in schema) should be configured to indicate which time period should be archived. In addition the workflow id (workflowId in schema) should be configured which should be archived.

Exporting the current forecast in a workflow which didn't computed the forecast

It is also possible to archive the current forecast to the archive from another workflow that computed the forecast.

If the export is done without an relative period but not in the workflow which created the forecast the current forecast will be exported. If you have used modifiers in this workflow you should also configure the relatedWorkflowId to indicate which workflow was run to compute the forecast, this will ensure that the modifiers used in this workflow will also be exported to the archive.

The associated root directory structure of the Delft-FEWS export for this type of dataset is as follows:
<archiveRoot>/<yyyy>/<MM>/<areaId>/<dd>/simulated/<workflowId><*TimeZero*><DispatchTime>
Where <dd> refers to the date of the forecast time T0.
This directory holds the metaData.xml as well as runInfo.xml file with the FEWS taskrun properties (see Figure 3.8). Within this directory the following sub-folders may exist:
/timeseries, /reports, /modifiers, /states
The exportArchiveModule.xsd has a dedicated exportSimulated section to configure the messages that need to be archived (see Figure 3.7). The associated specification is given in Table 4.4.
Table 4.4 Delft-FEWS export configuration for archiving simulations

Store time series in the netcdf storage

It is possible to store a simulation partly in the Open Archive (meta data, states, reports, modifiers and reports) and partly (time series) in the netcdf storage. This can be useful when you want to have a 'clean' time series only storage*. This can be configured by add the element netcdfStorageExport to the general section. An example is given below.*

```
<?xml version="1.0" encoding="UTF-8"?>
<exportArchiveModule 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/exportArchiveModule.
xsd">
        <exportSimulated>
                <general>
                        <archiveFolder>folder</archiveFolder>
                        <areaId>area</areaId>
                        <netcdfStorageExport>
                                <matchingAttributeId>attribute1</matchingAttributeId>
                                <matchingAttributeId>attribute2</matchingAttributeId>
                        </netcdfStorageExport>
                        <unitConversionsId>UnitConversionArchive</unitConversionsId>
                        <omitMissingValues>true</omitMissingValues>
                        <coldStartDataExportRelativeStartTime value="-1" unit="day"><
/coldStartDataExportRelativeStartTime>   </general>
                <activities>
                        <netcdfExportActivities>
                                <netcdfExportActivity>
                                        <fileName>test.nc</fileName>
                                        <timeSeriesSet>
                                                <moduleInstanceId>test</moduleInstanceId>
                                                <valueType>scalar</valueType>
                                                <parameterId>par</parameterId>
                                                <locationId>loc</locationId>
                                                <timeSeriesType>simulated forecasting</timeSeriesType>
                                                <timeStep unit="hour"></timeStep>
                                                <readWriteMode>add originals</readWriteMode>
                                        </timeSeriesSet>
                                </netcdfExportActivity>
                        </netcdfExportActivities>
                </activities>
        </exportSimulated>
</exportArchiveModule>
```

Because the time series and the meta data for the simulation are now stored separately it is important to know to which simulation a set of netcdf files belong. The meta data and the netcdf files are matched by using the task run id. In addition they are matched by the configured matchingAttributeId. The meta data of a simulation has the value of these attributes stored. They need to match with the values in the actual netcdf files. In netcdf storage you need to configure were the time series are stored so that they can be harvested by the harvester of the external netcdf storage. Below a configuration example is shown.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSpy v2014 rel. 2 sp1 (http://www.altova.com) by Afdeling ICT (Stichting Deltares) -->
<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="rwsos.noordzee.matroos.maps">
                <dataFolder>c:\FEWS\archive\matroos\maps</dataFolder>
                <matchingAttributeId>source</matchingAttributeId>
                <matchingAttributeId>issue_time</matchingAttributeId>
        </simulatedForecastingNetcdfStorage>
</externalStorage>
```

Element	Format	Description
General ExportForecast Section ComplexType		
archiveFolder	string	Export destination folder, assumes that the account running the FEWS (FSS) application has write access
relativePeriod		Exports entire simulated timeseries by workflow. If no relativePeriod is specified the Current simulated forecast is exported
workflowId	string	This option should be in combination with the relativePeriod. Al the forecasts of the workflowId which exists in the relative period will be exported
idMap	string	idMap applied to translate internal FEWS identifiers to identifiers that meet NetCDF-CF criteria.E.g. netcdf does not allow a full stop ('.') in the variable name
unitConversionsId	string	Available since 2022.02. Optional. Id of UnitConversions to be used for unit mapping.
omitMissingValues	boolean	Available since 2022.02. Optional. If set to true, time steps with missing values won't be written to the export file.
coldStartDataExportRelativeStartTime		Available since 2022.02. Optional. If this is specified, then the period for which data is exported will be truncated for simulated data that was created in a run that started with a cold state. In other words for this type of data, any data before the specified start time will not be exported. This start time is relative to the time zero of the run in which the data was created. This can be used to avoid exporting data that was created during the warm-up period of a model run after a cold start. Note: this option only works when the default state selection was configured to be cold state for the run in which the data was created.
relatedWorkflowId	string	If the export is done without the relativePeriod and workflowId option but not in the workflow which created the forecasts the workflow id of the forecast you are trying to export should be configured to make sure that the modifiers which are used in this forecast will be exported.
ForecastActivities ComplexType		

NetcdfForecastExportActivitiesComplexType		
NetcdfForecastExportActivity ComplexType*		
fileName	string	without nc extension, preferably no spaces
areaId	string	area to which the dataset belongs
ncMetaData	string elem.	optional metadata tags within NetCDF file following CF convention. Supported by the internal catalogue of the THREDDS Data Server
includeFlags	bool	only applied for scalar values. default=TRUE; if TRUE, a list of flags is stored, with each value pointing to the associated flag
includeComments	bool	only applied for scalar values. default=TRUE; if TRUE, a list of comments is stored, with each value pointing to the associated comment
timeSeriesSets		FEWS timeseries sets
ReportsExportActivity ComplexType*		
subfolder	string	Subdirectory within the 'reports' directory
moduleInstanceId	string	moduleInstanceId which created the report
ModuleStatesExportActivity ComplexType*		
moduleInstanceId	string	moduleInstanceId which created the state

*When an existing file is locked while it needs to be overwritten, the export function writes a new temporary file. The FileSweeper, a scheduled process, renames this file when the lock is removed from the original file.

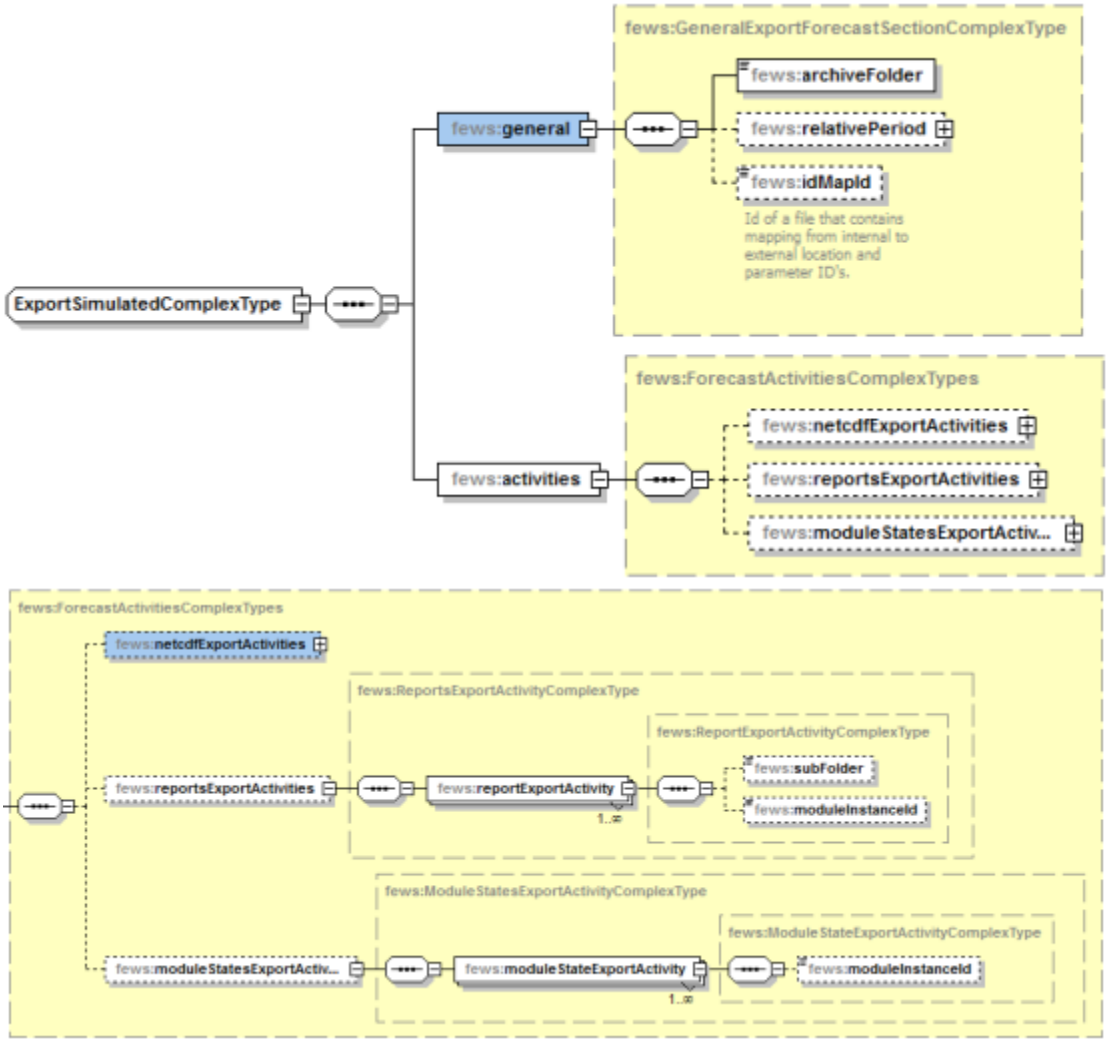


Figure 4.5 Delft-FEWS export configuration for archiving simulations

Configuration archiving by Delft-FEWS

Delft-FEWS can export the current configuration to the archive via the ArchiveExportModule (exportConfig activity). The configuration thus is exported as part of the workflow. Before exporting, FEWS checks if the config of the current revisionId was already written on the file system. If the files are moved from the exported directory, it will thus export again later, using the same file and folder names, so that in a regular data transfer setup the files will get be overwritten in each export again.

The associated root directory structure of the Delft-FEWS export for this type of dataset is as follows:
<archiveRoot>/data/config/<areaId>/<yyyymmdd>/<mastercontroller_id>_<revisionId>

The date refers to the revision date of the configuration. The revisionId is used in the subfolder name (see path above) and the file name of the z.ip file, i. e.: <mastercontroller_id>_<revisionId>.zip. The .zip file will be accompanied by a suiting metaData.xml file.
The exportArchiveModule.xsd has a dedicated exportConfig section to setup the export of the Configuration (see Figure 4.6).

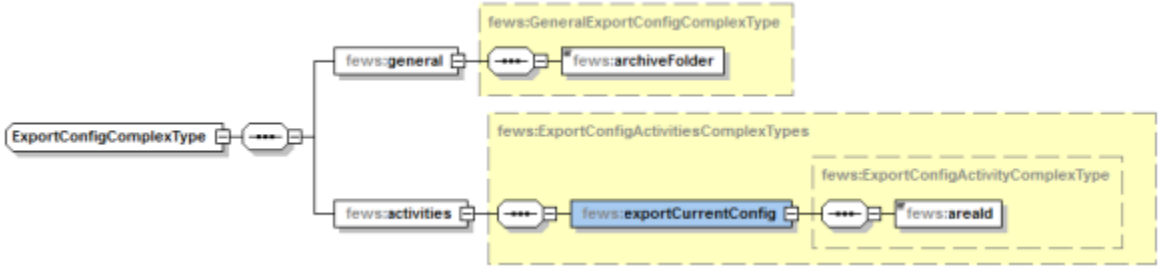


Figure 4.6 Delft-FEWS export configuration for archiving configurations

Table 4.5 Delft-FEWS export configuration for archiving a configuration

Element	Format	Description
GeneralExportConfig ComplexType		
archiveFolder	string	Export destination folder, assumes that the account running the FEWS (FSS) application has write access
ExportConfigActivities ComplexType		
ExportConfigActivity ComplexType*		
areaId	string	areaId to which the dataset belongs

Archiving rating curves by Delft-FEWS

Delft-FEWS can export rating curves to the archive via the ArchiveExportModule (exportRatingCurves activity). The entire history of the rating curves is exported and the export format is PI-XML (not NetCDF!). The associated root directory structure of the Delft-FEWS export for this type of dataset is as follows *FEWS-10198: update directory structure to above setup*:
<archiveRoot>/ratingcurves/<areaId>
The configurator can choose between exporting the full set or just the rating curves that have changed, with or without modifier changes.
The exportArchiveModule.xsd has a dedicated exportRatingCurve section to setup the export of the Configuration. Table 4.6 and Figure 4.7 discuss the general section.

Table 4.6 Delft-FEWS export configuration for archiving rating curves (general section)

Element	Format	Description
GeneralExportRatingCurves ComplexType		
archiveFolder	string	Export destination folder, assumes that the account running the FEWS (FSS) application has write access
relativePeriod		If defined, only rating curves which have database changes during the relativePeriod (relative to T0) are exported. If no relativePeriod is specified all available rating curves are exported
Prefix-timeZeroFormattingString	string	Adds T0 stamp as prefix to file name E.g.yyyymmdd_hhmm
idMap	string	idMap applied to translate internal FEWS identifiers to external identifiers
unitConversionsId	string	Allows conversion of rating table units
includeModifiers	bool	Default=FALSE; If TRUE, rating curve modifiers should be included in the export

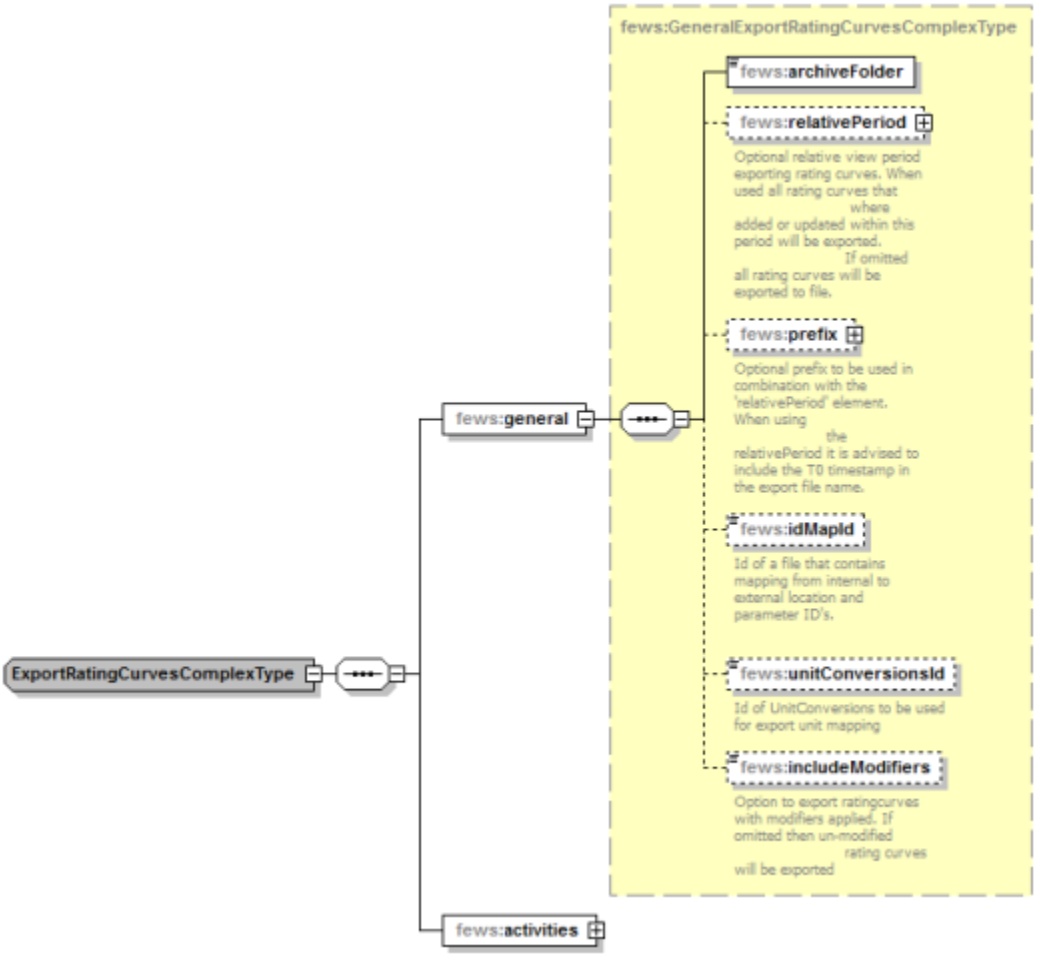


Figure 4.7 Delft-FEWS export configuration for archiving rating curves, general section

Table 4.7and Figure 4.8 discuss the activities section of for exporting rating curves.

Table 4.7 Delft-FEWS export configuration for archiving rating curves (activities section)

Element	Format	Description
RatingCurvesActivities ComplexType		
RatingCurvesExportActivity ComplexType		
fileName	string	without extension, preferably no spaces
areaId	string	area to which the dataset belongs
linearTableStageresolution	string	Stage increment between the rows in a rating table
locationId/locationSetId	string	Locations for which rating curves should be exported to this file

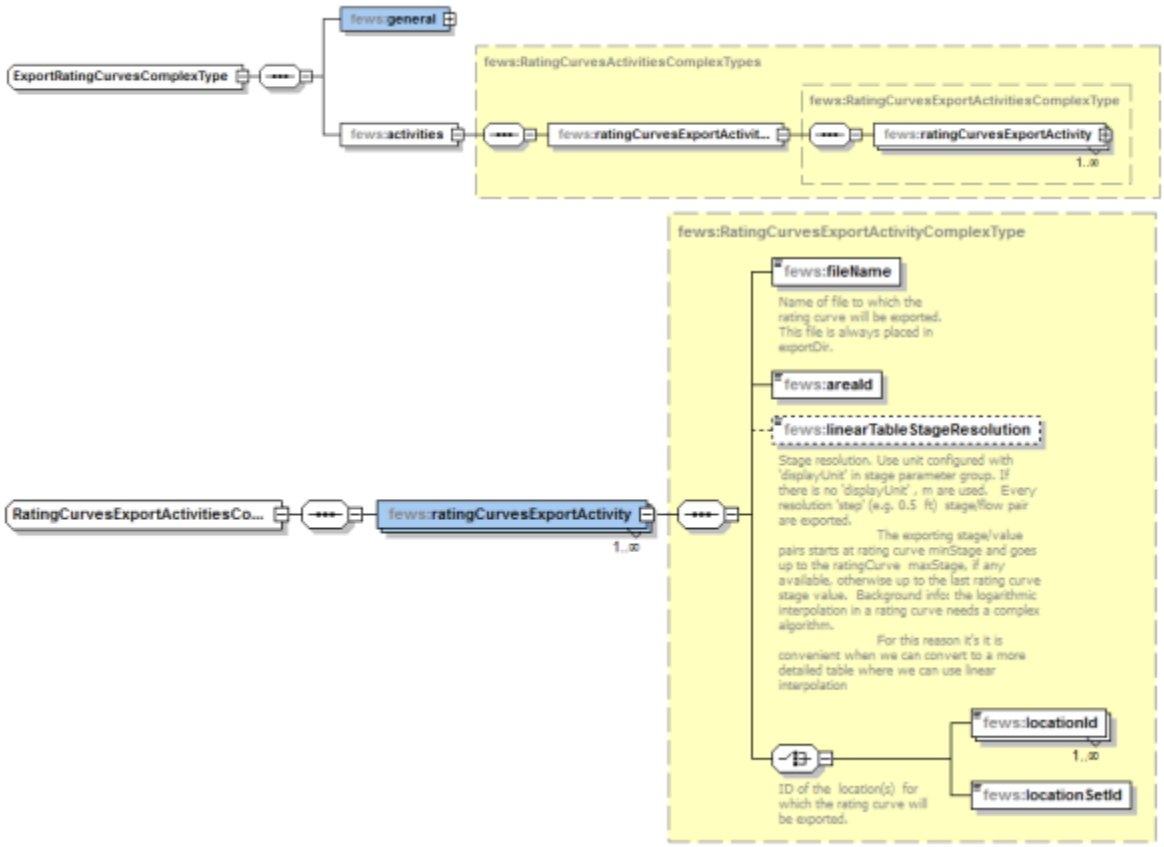


Figure 4.8 Delft-FEWS export configuration for archiving rating curves, activities section

Archiving Delft-FEWS snap shots

Location of the snapshot

Snap shots of the Delft-FEWS database can be archived via the archiveExportModule (exportSnapShot activity). The snapshot is by definition a firebird localDataStore.
For snapshots a dataset is generated and stored under the configured area at the moment of workflow execution. The associated directory structure of the Delft-FEWS export for this type of dataset is as follows:
<archiveRoot>/<yyyy>/<MM>/<areaId/<snapshot>/
Figure 4.9 and Table 4.8 documents the associated elements in the exportSnapShot activity.

Incremental export

Since 2019.02 the archive snapshot export is incremental. A copy of the last exported snapshot is made to use as base for the new export. The modified and new rows are copied. Rows that are no longer in the specified profile or central database are deleted from the copy. The snapshot is compacted afterwards . Whether or not a previous snapshot is used as base, the created snapshot contains the same rows and has the same size.

Delft-FEWS stores the location of the last snapshot in a text file: $ARCHIVE_FOLDER$\lastExportedSnapshotPath_$AREA$.txt, If this file does not exist, Delft-FEWS will export a complete snapshot.

Table 4.8 Delft-FEWS export configuration for archiving database snap shots

Element	Format	Description
general ComplexType		
archiveFolder	string	Export destination folder, assumes that the account running the FEWS (FSS) application has write access
zipExportedSnapShot	boolean	Place local datastore in zip file. Default true
singularExport		The archive folder structure will be ignored and the snapshot will be written directly in the archiveFolder

ExportSnapShotActivity ComplexType		
areaId	string	area (folder) where snapshot is archived

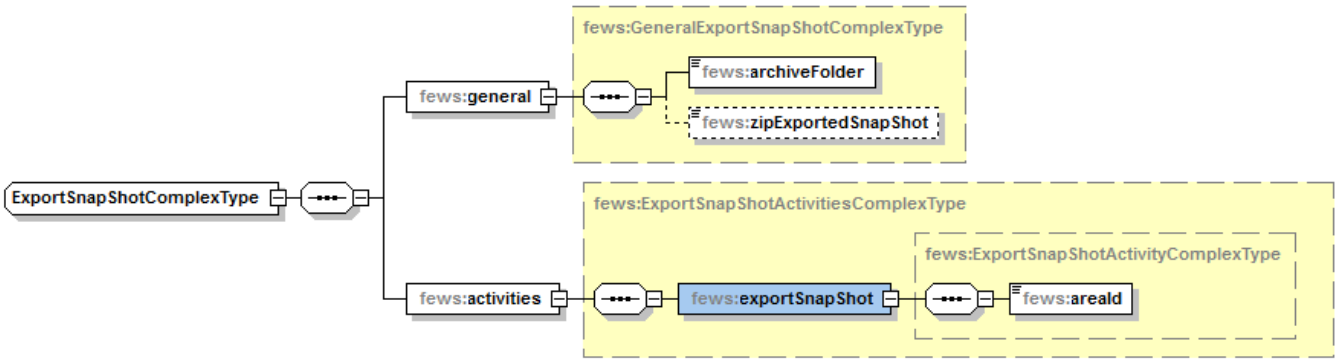


Figure 4.9 Delft-FEWS export configuration for archiving database snap shots

```
<exportArchiveModule xsi:schemaLocation="http://www.wldelft.nl/fews http://fews.wldelft.nl/schemas/version1.0
/exportArchiveModule.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.wldelft.nl
/fews">
    <exportSnapShot>
        <general>
            <archiveFolder>$ARCHIVE_DIR$</archiveFolder>
        </general>
        <activities>
            <exportSnapShot>
                <areaId>area</areaId>
                <filter id="only time series">
                    <xmlConfig enabled="false" name="Default xml config" synchLevel="11"/>
                    <coldStates enabled="false" name="Default cold states" synchLevel="11"/>
                    <moduleDataSets enabled="false" name="Default module data sets" synchLevel="11"/>
                    <mapLayers enabled="false" name="Default map layers" synchLevel="11"/>
                    <icons enabled="false" name="Default icons" synchLevel="11"/>
                    <reportTemplates enabled="false" name="Default report templates" synchLevel="11"/>
                    <reportImages enabled="false" name="Default report images" synchLevel="11"/>
                    <continuousTimeSeries enabled="true" name="Simulated" synchLevel="0" maxAge="1000" unit="
week"/>
                    <continuousTimeSeries enabled="true" name="Telemetry" synchLevel="1" maxAge="1000" unit="
week"/>
                    <continuousTimeSeries enabled="true" name="Manual" synchLevel="5" maxAge="1000" unit="week"
/>
                    <continuousTimeSeries enabled="true" name="Astronomical and climatological" synchLevel="4"
maxAge="1000" unit="week"/>
                    <continuousTimeSeries enabled="true" name="Small external forecast grids" synchLevel="6"
maxAge="1000" unit="week"/>
                    <continuousTimeSeries enabled="true" name="Large external forecast grids" synchLevel="16"
maxAge="10000" unit="week"/>
                    <warmStates enabled="false" name="Warm states" maxAge="10" unit="week"/>
                    <logEntries enabled="false" name="Log Entries" maxAge="1" unit="week"/>
                    <thresholdEvents enabled="false" name="Threshold Events" maxAge="1" unit="week"/>
                </filter>
            </exportSnapShot>
        </activities>
    </exportSnapShot>
</exportArchiveModule>
```

The base builds and non-default rootConfig files are always excluded from the snapshot

Archiving Delft-FEWS products

The Open Archive offers the possibility to archive products. A product is a collections of files which are stored in the archive. This option can be used to archive for example state files, reports or any other kind of file.

A product can be labelled with an area id and a source id.

It is possbile to archive products in the following ways:

1. By defining which files should be archived. In this option it is possible to use wildcards and it is also possible to zip some files into a zipfile
2. By defining an import folder in combination with a file pattern. Only the files which are found in this import folder and comply to this file pattern will be archived. The pattern will also be used to identify the creation time of the product.

In the sections below both options will be explained in more detail

Archiving files by defining which files should be archived

It is possible to define individual files which should be archived as part of the product. But it is also possible to archive files based on a wild card.

Table 4.8 Delft-FEWS export configuration for archiving database snap shots

Element	**Format**	**Description**
general ComplexType		
archiveFolder		
ExportProductActivity ComplexType		
areaId	string	Id of the area to which this product belongs
sourceId	string	Id of the source to which this product belongs
sourceFile	string	File which should be archived as part of the product
moveFile	boolean	If this option is set to true the source file will be deleted after it has been archived.
sources/zipProduct	boolean	if this option is set to true the files will
sources/sourceFiles	string	
sources/relativePath	string	the files identified by the tag sourceFiles will be exported to folder with the configurewithin the main folder of the data set identfied
moveFile	boolean	If this option is set to true then the source file will be deleted after the file has been copied to the archive

An configuration example is given below.

```
<exportArchiveModule
      xsi:schemaLocation="http://www.wldelft.nl/fews http://fews.wldelft.nl/schemas/version1.0
/exportArchiveModule.xsd"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.wldelft.nl/fews">
   <exportProducts>
      <general>
         <archiveFolder>$ARCHIVE_DIR$</archiveFolder>
      </general>
      <activities>
         <exportProduct>
            <areaId>area</areaId>
            <sourceId>source</sourceId>
            <sourceFile>$INPUT_FOLDER$/sourceA</sourceFile>
            <sourceFile>$INPUT_FOLDER$/sourceB</sourceFile>
            <moveFile>true</moveFile>
         </exportProduct>
      </activities>
   </exportProducts>
</exportArchiveModule>
```

An configuration example is given below.

```
<exportArchiveModule
      xsi:schemaLocation="http://www.wldelft.nl/fews http://fews.wldelft.nl/schemas/version1.0
/exportArchiveModule.xsd"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.wldelft.nl/fews">
   <exportProducts>
      <general>
         <archiveFolder>$ARCHIVE_DIR$</archiveFolder>
      </general>
      <activities>
         <exportProduct>
            <areaId>areaTest</areaId>
            <sourceId>sourceTest</sourceId>
            <sources>
               <zipProduct>true</zipProduct>
               <sourceFiles>$INPUT_FOLDER$/testdata/exportandmoveandzip/source?</sourceFiles>
               <relativePath>aa</relativePath>
            </sources>
            <moveFile>true</moveFile>
         </exportProduct>
      </activities>
   </exportProducts>
</exportArchiveModule>
```

Archiving products by defining an import folder and a date/time pattern

It is possible to archive products from an import folder. The import folder should contain the files which should be exported to the archive.

After the files are archived they will be deleted from the import folder. Each file will be archived as a separate data set in the archive.

The creation time of the product will be derived from the file name by using a file pattern.

There are two options for deriving the creation time from the file name:

1. Derive the creation time from a fixed file name pattern. This can be configured by using the option fileNameProductDateTimePattern:
2. Derive the creation time from the postfix of the file name. This can be configured by using the option productFileNamePostFixDateTimePattern

If a file doesn''t comply to the configured file pattern then the file will not be archived.

An configuration example is given below

```
<exportArchiveModule
      xsi:schemaLocation="http://www.wldelft.nl/fews http://fews.wldelft.nl/schemas/version1.0
/exportArchiveModule.xsd"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.wldelft.nl/fews">
   <exportProducts>
      <general>
         <archiveFolder>$ARCHIVE_DIR$</archiveFolder>
      </general>
      <activities>
         <exportProduct>
            <areaId>areaTest</areaId>
            <sourceId>sourceTest</sourceId>
            <importFolder>$IMPORT_DIR$</importFolder>
            <fileNameProductDateTimePattern>yyyyMMdd'.nc'</fileNameProductDateTimePattern>
         </exportProduct>
      </activities>
   </exportProducts>
</exportArchiveModule>
```

Archiving to Archive Database

Archiving observed data to the archive database

Below an example of how to configure an export of scalar data to the archive database.

```
<exportArchiveModule xsi:schemaLocation="http://www.wldelft.nl/fews file:///c:/fews_trunk/xml-schemas
/exportArchiveModule.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                                    xmlns="http://www.wldelft.nl/fews">
        <exportExternalHistoricalTimeSeriesToArchiveDatabase>
                <general>
                        <relativePeriod start="-10" end="0" unit="day"/>
                        <exportUnitConversionsId>UnitConversionDatabase</exportUnitConversionsId>
                        <timeZoneName>CET</timeZoneName>
                        <options>
                                <double key="doubleKey" value="12"/>
                        </options>
                </general>
                <activities>
                        <areaId>myAreaId</areaId>
                        <sourceId>mySourceId</sourceId>
                        <timeSeriesSet>
                                <moduleInstanceId>ArchiveExportObservedToArchiveDatabaseTest</moduleInstanceId>
                                <valueType>scalar</valueType>
                                <parameterId>H.m</parameterId>
                                <locationId>H-2001</locationId>
                                <locationId>H-2002</locationId>
                                <timeSeriesType>external historical</timeSeriesType>
                                <timeStep unit="hour"/>
                                <readWriteMode>add originals</readWriteMode>
                        </timeSeriesSet>
                </activities>
        </exportExternalHistoricalTimeSeriesToArchiveDatabase>
</exportArchiveModule>
```

The general section can be used to define generic options like the period to export. The MongoDB-plugin has the option to store data in GMT and in a local time zone. To configure the local time zone the timeZoneName can be used. The options tag can be used to define custom properties which be send the plugin. This makes it possible to configure custom properties for each database plugin.

The activities section is used to define the area id and the source id. Both are used by the archive display to facilitate searching for data. In the example above a time series set is configured. It is also possible to use a time series filter. If this is used then all the time series of a certain workflow which comply to that filter will be exported. If no workflow is configured then the current workflow is used otherwise the current (or last) run of the configured workflow is used.

```
<exportArchiveModule xsi:schemaLocation="http://www.wldelft.nl/fews file:///c:/fews_trunk/xml-schemas
/exportArchiveModule.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                                        xmlns="http://www.wldelft.nl/fews">
        <exportExternalHistoricalTimeSeriesToArchiveDatabase>
                <general>
                        <relativePeriod start="-10" end="0" unit="day"/>
                        <exportUnitConversionsId>UnitConversionDatabase</exportUnitConversionsId>
                        <timeZoneName>CET</timeZoneName>
                        <options>
                                <double key="doubleKey" value="12"/>
                        </options>
                </general>
                <activities>
                        <areaId>myAreaId</areaId>
                        <sourceId>mySourceId</sourceId>
                        <workflowTimeSeriesSelection>
                                <timeSeriesFilter>
                                        <parameterId>H.m</parameterId>
                                </timeSeriesFilter>
                        </workflowTimeSeriesSelection>
                </activities>
        </exportExternalHistoricalTimeSeriesToArchiveDatabase>
</exportArchiveModule>
```

Archiving simulated data to the archive database

Below an example of how to configure an archive export of simulated data to the archive database.

```
<exportArchiveModule xsi:schemaLocation="http://www.wldelft.nl/fews file:///c:/fews_trunk/xml-schemas
/exportArchiveModule.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                                    xmlns="http://www.wldelft.nl/fews">
        <exportSimulatedTimeSeriesToArchiveDatabase>
                <general>
                        <exportUnitConversionsId>UnitConversionDatabase</exportUnitConversionsId>
                        <timeZoneName>CET</timeZoneName>
                        <options>
                                <double key="doubleKey" value="12"/>
                        </options>
                </general>
                <activities>
                        <areaId>myAreaId</areaId>
                        <sourceId>mySourceId</sourceId>
                        <timeSeriesSet>
                                <moduleInstanceId>ArchiveExportSimulatedToArchiveDatabaseTest</moduleInstanceId>
                                <valueType>scalar</valueType>
                                <parameterId>H.m</parameterId>
                                <locationId>H-2001</locationId>
                                <locationId>H-2002</locationId>
                                <timeSeriesType>simulated forecasting</timeSeriesType>
                                <timeStep unit="hour"/>
                                <readWriteMode>read complete forecast</readWriteMode>
                        </timeSeriesSet>
                </activities>
        </exportSimulatedTimeSeriesToArchiveDatabase>
        <exportSimulated>
                <general>
                        <archiveFolder>$ARCHIVE_DIR$</archiveFolder>
                        <areaId>area</areaId>
                        <sourceId>source</sourceId>
                </general>
                <activities></activities>
        </exportSimulated>
</exportArchiveModule>
```

At the moment it is only possible to export the current simulated data to the archive database. The options in the general section are similar to the options in the observations export. The activities section can be used to define which time series should be exported. This can be done by using time series sets or by using a time series filter.

Below an example of how to configure an export with time series filters.

```
<exportArchiveModule xsi:schemaLocation="http://www.wldelft.nl/fews file:///c:/fews_trunk/xml-schemas
/exportArchiveModule.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                                    xmlns="http://www.wldelft.nl/fews">
        <exportSimulatedTimeSeriesToArchiveDatabase>
                <general>
                        <exportUnitConversionsId>UnitConversionDatabase</exportUnitConversionsId>
                        <timeZoneName>CET</timeZoneName>
                        <options>
                                <double key="doubleKey" value="12"/>
                        </options>
                </general>
                <activities>
                        <areaId>myAreaId</areaId>
                        <sourceId>mySourceId</sourceId>
                        <workflowTimeSeriesSelection>
                                <timeSeriesFilter>
                                        <parameterId>H.m</parameterId>
                                </timeSeriesFilter>
                        </workflowTimeSeriesSelection>
                </activities>
        </exportSimulatedTimeSeriesToArchiveDatabase>
</exportArchiveModule>
```

Archiving external forecast data to the archive database

Below an example of how to configure an archive export of external forecasts to the archive database.

```
<exportArchiveModule xsi:schemaLocation="http://www.wldelft.nl/fews file:///c:/fews_trunk/xml-schemas
/exportArchiveModule.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                                        xmlns="http://www.wldelft.nl/fews">
        <exportExternalForecastingTimeSeriesToArchiveDatabase>
                <general>
                        <exportUnitConversionsId>UnitConversionDatabase</exportUnitConversionsId>
                        <timeZoneName>CET</timeZoneName>
                        <options>
                                <double key="doubleKey" value="12"/>
                        </options>
                </general>
                <activities>
                        <areaId>myAreaId</areaId>
                        <sourceId>mySourceId</sourceId>
                        <timeSeriesSet>
                                <moduleInstanceId>ArchiveExportExternalForecastToArchiveDatabaseTest<
/moduleInstanceId>
                                <valueType>scalar</valueType>
                                <parameterId>H.m</parameterId>
                                <locationId>H-2001</locationId>
                                <locationId>H-2002</locationId>
                                <timeSeriesType>external forecasting</timeSeriesType>
                                <timeStep unit="hour"/>
                                <readWriteMode>read complete forecast</readWriteMode>
                        </timeSeriesSet>
                </activities>
        </exportExternalForecastingTimeSeriesToArchiveDatabase>
</exportArchiveModule>
```

It is also possible to define which time series should be exported by using a time series filter. Below an example.

```
<exportArchiveModule xsi:schemaLocation="http://www.wldelft.nl/fews file:///c:/fews_trunk/xml-schemas
/exportArchiveModule.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                                        xmlns="http://www.wldelft.nl/fews">
        <exportExternalForecastingTimeSeriesToArchiveDatabase>
                <general>
                        <exportUnitConversionsId>UnitConversionDatabase</exportUnitConversionsId>
                        <timeZoneName>CET</timeZoneName>
                        <options>
                                <double key="doubleKey" value="12"/>
                        </options>
                </general>
                <activities>
                        <areaId>myAreaId</areaId>
                        <sourceId>mySourceId</sourceId>
                        <workflowTimeSeriesSelection>
                                <timeSeriesFilter>
                                        <parameterId>H.m</parameterId>
                                </timeSeriesFilter>
                        </workflowTimeSeriesSelection>
                </activities>
        </exportExternalForecastingTimeSeriesToArchiveDatabase>
</exportArchiveModule>
```