02 Data Handling in Delft-FEWS

Introduction

One of the most important properties of Delft-FEWS as a forecasting system is its ability to efficiently deal with large volumes of dynamic data. Dynamic data covers mainly time series data in various formats (scalar- 0D, vector - 1D, longitudinal profile - 1D and 2D, grid - 2D and 3D, and polygon data - 2D). Dynamic data also includes the management of model states produced by the system. As Delft-FEWS is not only used as operational forecasting system, but also as a system to run climate scenarios, the length of each time series can be up to 2,000,000,000 time steps. A thorough understanding of how Delft-FEWS handles dynamic data is fundamental to correctly configuring an operational system. Specific optimisations are available for each type of dynamic data. This chapter introduces the concept of a "Time Series Set". A Time Series Set is used to retrieve data from and submit data to the database.

Types of time series

External and Simulated time series

Time series are considered to be available from two sources. All time series sourced from external systems are considered "External". All time series produced by the forecasting system itself are considered "Simulated".

Forecast and Historical time series

Time series are considered to be of two categories in relation to time. Historical time series are continuous time series that describe a parameter at a location over a period of time. Forecast time series are different to historical time series in that for each location and parameter one forecast is independent of another forecast. A forecast is characterised by its start time and the period it covers. Generally, when a new forecast is available for a given location and parameter, it will supersede any previous forecast for that location and parameter. Each forecast is therefore an independent entity.

On the basis of this, six categories of time series are identified;

+ External Historical
+ External Forecasting
+ Simulated Historical
+ Simulated Forecasting
+ Temporary
+ Temporary External Forecasting (since 2017.01)

There are significant differences in how each of these time series are handled in Delft-FEWS.

External Historical time series

In an operational system, Delft-FEWS incrementally imports observed data as it becomes available from external systems. These data should be imported as External Historical time series. When data marked as external historical is presented to the system with exactly the same values and covering the same period as data for that location/parameter already available in the database, then it will be ignored (i.e. not imported). Only new data are imported and stored. If data for a given period is already available but is changed (manual edit or update), the new values will be added to the database. For each item of data added to the database, a time stamp is included to specify when the data was made available to the system.

When external historical data are requested from the database, the most recently available data over that whole period is returned. If the data for that period was imported piecewise, the individual pieces will be merged prior to the data being returned. An example is given in Figure 1, where data are imported sequentially. Each dataset imported/edited is indicated using a different line style. When the complete series (a) is requested, the most recent data available over the complete period is merged and returned. The data imported at 12:00 partially overlaps that imported at 10:00. Since the 12:00 data is the most recent, it will persist in the complete series. A manual edit (or interpolation) may be done to fill the gap in the data. This will be returned in a subsequent request for the complete series. Although a complete series is returned, the data is stored as it is imported, including a time stamp indicating when the import happened. If at a later stage the data directly preceding the manual edit is requested, then the additional data will not be included in the complete series.

Figure 1 Schematic representation of data imported as external historical

External Forecasting time series

External forecasts are imported by Delft-FEWS as they are made available by other, external forecasting systems. Each forecast is imported and stored individually. External forecasts are referenced by the start time of the forecast. When retrieving an external forecast time series from the database, the most recently available forecast, as indicated by the forecast start time, will be returned. The most recently available forecast is determined as the latest forecast with a start time earlier or equal to the start of the forecast to be made using Delft-FEWS (forecast T0). It is thus not possible to see an external forecast time series on request, as the latest available is always returned.

With possible exceptions for modules considering multiple forecasts (e.g. performance module), only one external forecast is returned. Different external forecasts are not merged.

Simulated Historical time series

Simulated historical time series are similar to the external historical time series in that they are continuous in time. The difference is that the time series are referenced through the forecast (model) run they have been produced by. As a consequence the time series can be retrieved either by directly requesting it through opening the run and viewing, or if the run is approved. If you use an extended relativeViewPeriod and the readWriteMode "read only" with a simulated historical time series, you can access the combined results of several model runs (within the specified relativeViewPeriod), similar to the default behavior of the merged external historical time series. If you use readWriteMode 'read complete forecast' without a relativeViewPeriod, you will only obtain the current forecast.

Simulated historical time series are generally produced by model runs where a model initial state is used. Each time series has a history, i.e. the state used as its initial condition. Each state again has a history, i.e. the model run that produced the state. This history is used by the database in constructing a continuous time series.

Simulated Forecasting time series

Simulated forecast time series are again similar to external forecasting time series. Again the main differences is that they are referenced through the forecast (model) run they have been produced by. As a consequence the time series can be retrieved either by directly requesting it through opening the run and viewing, or if the run is approved. Simulated forecast time series are treated in the same way as the external forecast time series in that the last approved forecast (referred to as the current forecast) is seen as a default. All other runs can be seen on request only. Note that the last approved forecast which is shown by default may not be the last available forecast.

Figure2 schematically shows how a sequence of runs producing simulated historical and simulated forecasting time series are stored. Each simulated historical run uses the module state saved at the end of the previous run. It can be seen that these simulated historical traces are treated as a continuous time series when requested later. For the forecasting time series, only the most recent (approved) time series is displayed.

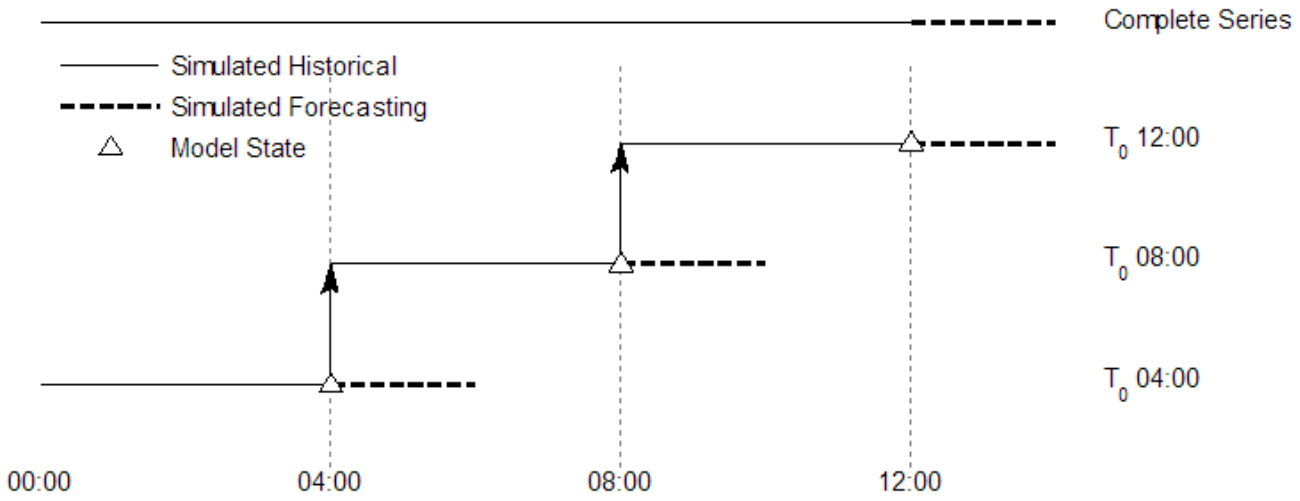


Figure 2 Schematic overview of handling simulated forecasting and simulated historical time series. Three subsequent forecasts are shown, and the resulting complete time series returned when requested after 12:00. The historical time series is traced back using the state used to create the link to a previous run. For the forecast time series the most recent forecast supersedes previous forecasts.

> The time series type simulated historical should only be assigned to time series that have a relation to a previous time series through a model state. In all other cases, the time series is independent, and should be allocated simulated forecasting as time series type.

Temporary time series

Temporary time series are deleted at the end of a run. They are only visible for the running task run.

Temporary External Forecasting time series (since 2017.01)

This replaces an external forecast with synch level 9 with a parameter that is only used in combination with synch level 9 and not in the final result. Temporary time series are deleted at the end of a run. They are only visible for the running task run. You can now use the same parameter as used in the final result

Time Series Sets

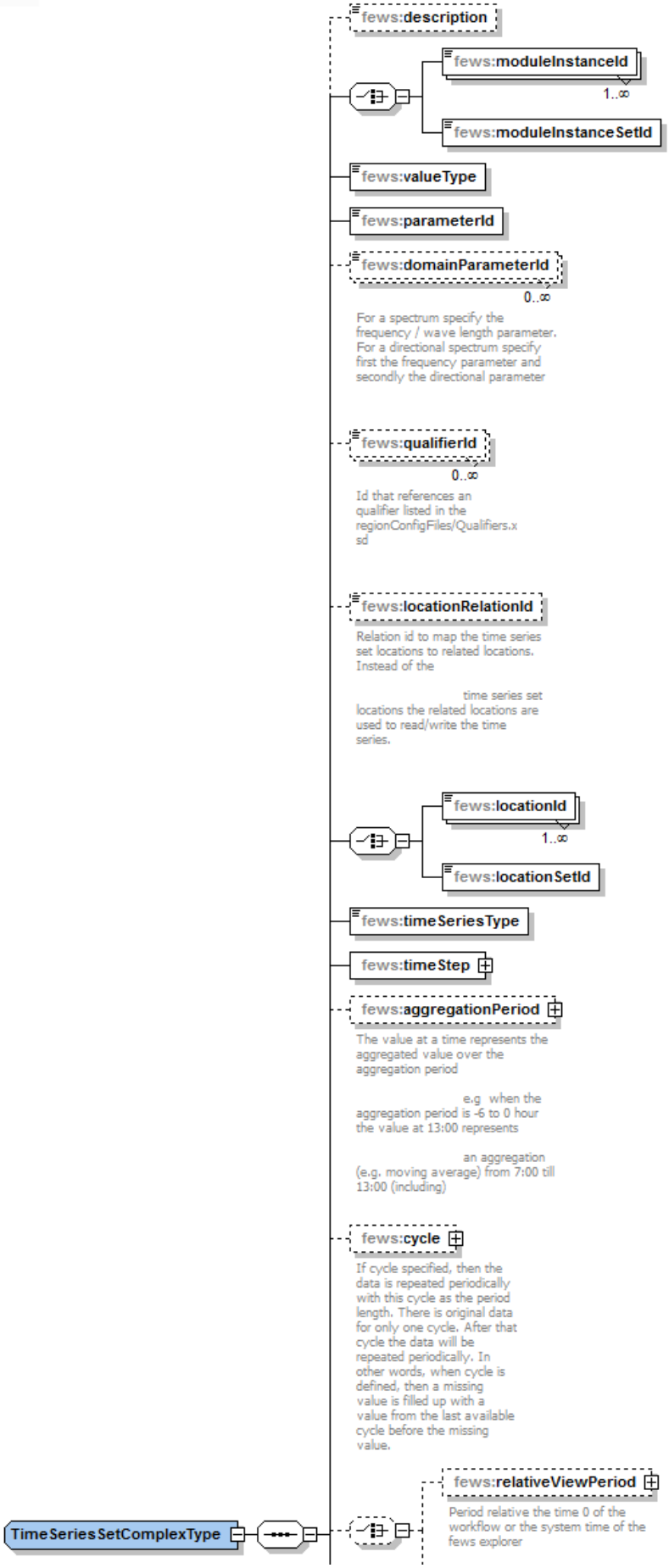
Any module in Delft-FEWS that requires data from the database, or produces data that must be stored in the database, does so through the use of a complex data type referred to as the Time Series Set. A time series set can be compared to a query that is run against the database. It contains all the keys to uniquely identify the set of data to be retrieved (for more information on key attributes, see Key attributes).

Time series sets form a large part of the configuration. Most modules have a standard structure, where the configuration starts with a request of specific set of data from the database using one or more input time series sets, a number of functional items which describe how the data is transformed, and one or more output time series sets which are used to store the data in the database under a unique combination of keys.

Figure3 shows the elements of the Time Series Set complex type. a number of these elements are compulsory (solid borders), while other elements are optional (dashed borders). If any of the required elements is omitted then the primary validation of that configuration will fail.

Depending on the information required, each of the elements will be used differently. Some elements are simple flags, indicating specific properties of the time series set if they are present. For others a string or value must be given in the configuration. In code this will mean that value or string is assigned to a variable with the name of the element. Other elements may also contain properties. The example of the time series set below illustrates the different types of element of the time series set.

Optional items may also need to be required to fulfill the requirements of the module using the time series set. This will be indicated in this manual for those modules where appropriate.


fews:description
fews:moduleInstanceId
1..∞
fews:moduleInstanceSetId
fews:valueType
fews:parameterId
fews:domainParameterId
0..∞
For a spectrum specify the frequency / wave length parameter. For a directional spectrum specify first the frequency parameter and secondly the directional parameter
fews:qualifierId
0..∞
Id that references an qualifier listed in the regionConfigFiles/Qualifiers.xsd
fews:locationRelationId
Relation id to map the time series set locations to related locations. Instead of the
time series set locations the related locations are used to read/write the time series.
fews:locationId
1..∞
fews:locationSetId
fews:timeSeriesType
fews:timeStep
fews:aggregationPeriod
The value at a time represents the aggregated value over the aggregation period
e.g when the aggregation period is -6 to 0 hour the value at 13:00 represents
an aggregation (e.g. moving average) from 7:00 till 13:00 (including)
fews:cycle
If cycle specified, then the data is repeated periodically with this cycle as the period length. There is original data for only one cycle. After that cycle the data will be repeated periodically. In other words, when cycle is defined, then a missing value is filled up with a value from the last available cycle before the missing value.
fews:relativeViewPeriod
Period relative the time 0 of the workflow or the system time of the fews explorer
TimeSeriesSetComplexType

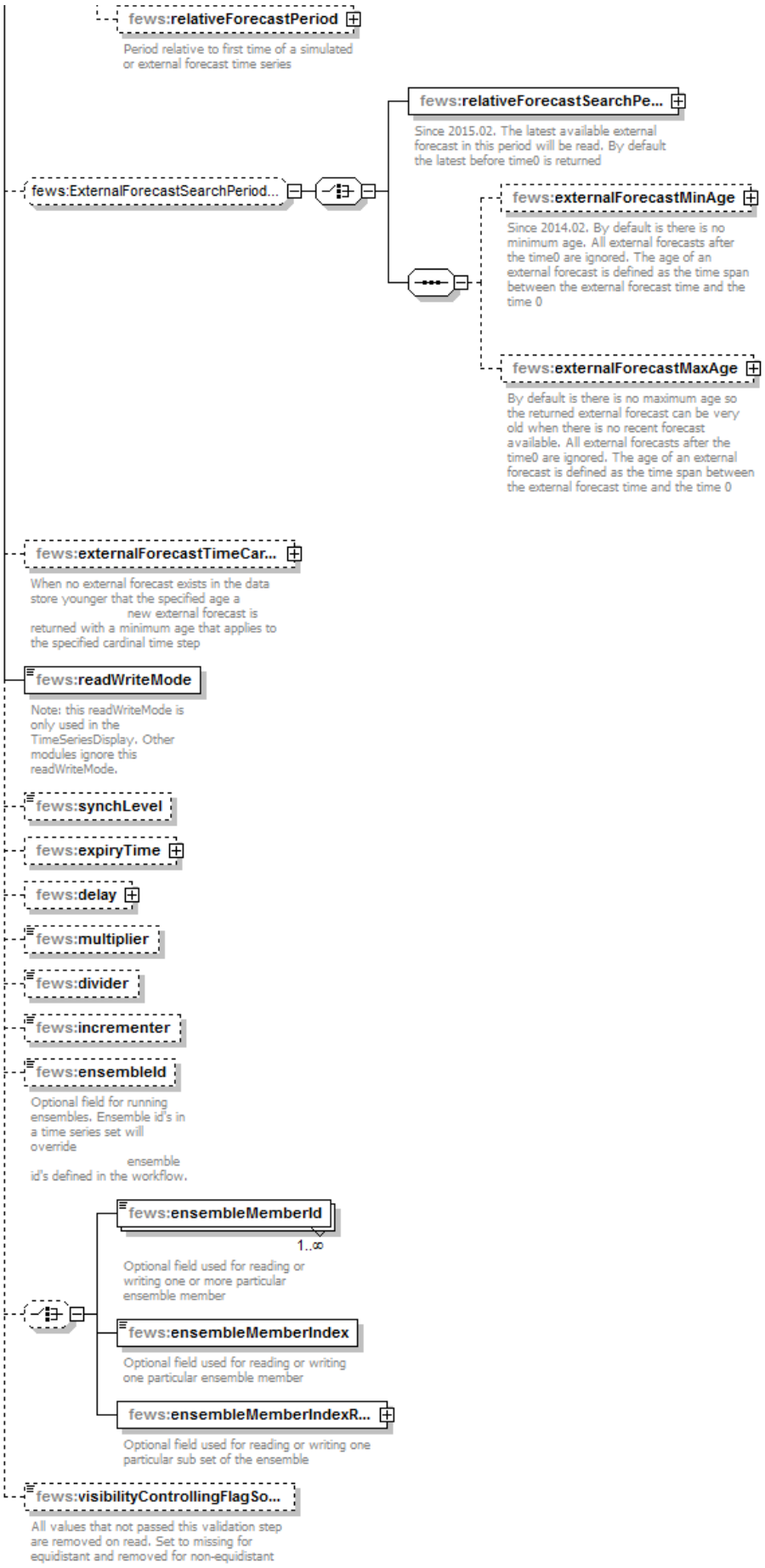

fews:relativeForecastPeriod
Period relative to first time of a simulated or external forecast time series
fews:relativeForecastSearchPe...
Since 2015.02. The latest available external forecast in this period will be read. By default the latest before time0 is returned
fews:ExternalForecastSearchPeriod...
fews:externalForecastMinAge
Since 2014.02. By default is there is no minimum age. All external forecasts after the time0 are ignored. The age of an external forecast is defined as the time span between the external forecast time and the time 0
fews:externalForecastMaxAge
By default is there is no maximum age so the returned external forecast can be very old when there is no recent forecast available. All external forecasts after the time0 are ignored. The age of an external forecast is defined as the time span between the external forecast time and the time 0
fews:externalForecastTimeCar...
When no external forecast exists in the data store younger that the specified age a new external forecast is returned with a minimum age that applies to the specified cardinal time step
fews:readWriteMode
Note: this readWriteMode is only used in the TimeSeriesDisplay. Other modules ignore this readWriteMode.
fews:synchLevel
fews:expiryTime
fews:delay
fews:multiplier
fews:divider
fews:incrementer
fews:ensembleId
Optional field for running ensembles. Ensemble id's in a time series set will override ensemble id's defined in the workflow.
fews:ensembleMemberId
1..∞
Optional field used for reading or writing one or more particular ensemble member
fews:ensembleMemberIndex
Optional field used for reading or writing one particular ensemble member
fews:ensembleMemberIndexR...
Optional field used for reading or writing one particular sub set of the ensemble
fews:visibilityControllingFlagSo...
All values that not passed this validation step are removed on read. Set to missing for equidistant and removed for non-equidistant

Figure3 Schema of the Time Series Set Complex type

description

This is an optional description for the time series set. It is only used a caption in configuration and is not stored with time series.

moduleInstanceId/moduleInstanceSetId

The module instance Id is the ID of the module that has written the data in the time series set to the database. This ID is one of the primary keys and is required to uniquely identify the data on retrieval.

In the time series set a single module instance Id may be referenced or multiple module instance Id's. The latter is done either by including a list of module instance Id's or by referencing a module instance set Id. This again resolves to a list of module instance set Id's as defined in the ModuleInstanceSets.xml configuration file.

One or more moduleInstanceId may be defined, or a single ModuleInstanceSetId. These cannot be mixed

valueType

This specifies the dimension/data type of the time series. This element is an enumeration of the next types;

- scalar
- longitudinalprofile
- grid
- polygon
- sample

parameterId

The parameterId describes the parameter of the data in the time series. This Id is a cross reference to the Parameters.xml configuration file in the regional configuration defining the parameters. The reference is not enforced through an enumeration in the XML schema. If a parameter not included in the parameter definition is referred to, an error will be generated and an appropriate message returned.

locationId/locationSetId

The locationId is a reference to the location for which the data series is valid. Each individual data series may belong to one location only. In the time series set a single location may be referenced or multiple locations may be referenced. The latter is done either by including a list of locationId's or by referencing a locationSetId. This again resolves to a list of locationId's as defined in the LocationSets.xml configuration file.

One or more locationId's may be defined, or a single locationSetId. These cannot be mixed.

timeSeriesType

This specifies the type of time series (see discussion above). This is an enumeration of;

- external historical
- external forecasting
- simulated historical
- simulated forecasting
- temporary (timeseries will not be stored in the database, but will be available for later processes in the same workflow)
- temporary external forecasting (same as previous, but has an externalForecastingtime)

timeStep

This is the time step of the time series. The time step can be either equidistant or non-equidistant. The time step is defined in the parameters of the timeStep element;

Attributes;

- **unit** (enumeration of: second, minute, hour, day, week, month, year, nonequidistant)
- **multiplier** defines the number of units given above in a time step (not relevant for nonequidistant time steps).
- **divider** same function as the multiplier, but defines fraction of units in time step.
- **timeZone** defines the timeZone of the timeStep, this is only relevant for units of a day or larger.

> ⚠ For hourly timesteps this may also be relevant in the case of half-hourly timezones. Untested at the moment.

For more information, have a look at the TimeStep Documentation

relativeViewPeriod / relativeForecastPeriod

The relative view period defines the span of time for which data is to be retrieved. This span of time is referenced to the start time of the forecast run (T0) the time series set is used in. If the time series set is not used in a forecast run (e.g. in the displays), then the reference is to the Delft-FEWS system time.

Parameters

- **unit** identifies the time unit with which the time span is defined (enumeration of second, minute, hour, day, week).
- **start** identifies the start time of the time span with reference to the T0 (in multiples of the unit defined).
- **end** identifies the start time of the time span with reference to the T0 (in multiples of the unit defined).
- **startOverrulable** Boolean flag to indicate if the start time given may be overruled by a user selection.
- **endOverrulable** Boolean flag to indicate if the end time given may be overruled by a user selection.

For equidistant time series, all values at the time interval within the span defined will be returned by the database following a request. If no values are found, then missing values will be returned at the expected time step. For non-equidistant time series, all values found within the time span are returned. If none are found, then no values are returned.

The relativeForecastPeriod Period is relative to the T0 of the current/selected forecast or the external forecast time of an (the latest) external forecast. Use relativeViewPeriod instead for simulated series created by the current task run.

> Of the parameters in the relative view period, only the end and unit parameters are compulsory. Generally, however, the start time will also be required. It is omitted only if the start time is determined through selection of a module state.

> This start and end time are the DEFAULT time span of data. Both can be overruled through user selection, there is no restriction on shortening the start time through user selection (since 2015.01).

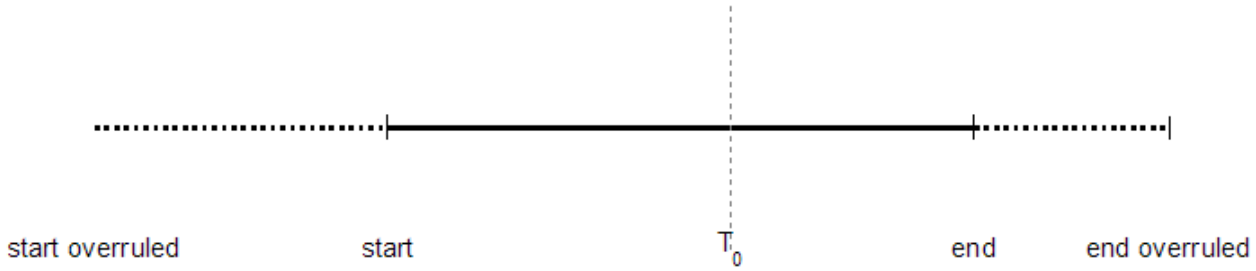


Figure 4 Schematic representation of the relative view period with reference to the T0. The start and end time defined may be overruled if the appropriate parameters are set to true.

cycle

If cycle is specified, the data is repeated periodically with this cycle as the period length. There is original data for only one cycle. After the dates specified in the cyclic time series the data will be repeated periodically. In other words, when cycle is defined, a missing value is filled up with a value from the last available cycle before the missing value. Be aware that retrieveing data from a cyclic timeseries requires a relative view period to ensure that the timestamps of the array are properly set. If you retrieve a cyclic timeseries with the 'read complete forecast' readWriteMode, the array read will have timestamps that correspond to the original data.

externalForecastMaxAge

when the externalForecastMaxAge is not configured there is no maximum age for a forecast series to be used, so the returned external forcast can be very old when there is no recent forecast available. ALL external forecasts after the T0 are ALWAYS ignored. The age of an external forecast is defined as the time span between the external forecast time and T0.

Attributes;

- **unit** (enumeration of: second, minute, hour, day, week)
- **multiplier** defines the number of units given above
- **divider** same function as the multiplier, but defines fraction of units.

externalForecastTimeCardinalTimeStep

When no external forecast exists in the data store younger than the specified age, a new external forecast is returned with a minimum age that applies to the specified cardinal time step.

Attributes;

- **unit** (enumeration of: second, minute, hour, day, week, nonequidistant)
- **multiplier** defines the number of units given above (not relevant for nonequidistant time steps).
- **divider** same function as the multiplier, but defines fraction of units.
- **timeZone** defines the timeZone, this is only relevant for units of a day or larger.

externalForecastSearchTimeStep

Since 2024.01. Only time series with an external forecast time that match this time step are visible while searching, e.g. an externalForecast timeseries that has a externalForecastTime of 18:00 in the GMT timeZone.

All timestep attributes are available, probably most relevant options are;

- **id** Id of the time step. You can reference time steps defined in the regionConfig/timeSteps.xml.
- **times** Defines the time step by a list of times without dates, e.g. "10:00 23:00"
- **timeZone** defines the timeZone, this is only relevant for units of a day or larger.

```
<timeSeriesSet>
        <moduleInstanceId>ImportACCESS-GE</moduleInstanceId>
        <valueType>grid</valueType>
        <parameterId>P.nwp.fcst</parameterId>
        <locationId>ACCESS-GE_grid</locationId>
        <timeSeriesType>external forecasting</timeSeriesType>
        <timeStep unit="hour"/>
        <relativeForecastPeriod unit="hour" start="0" end="246"/>
        <externalForecastSearchTimeStep times="18:00" timeZone="GMT"/>
        <readWriteMode>read complete forecast</readWriteMode>
</timeSeriesSet>
```

readWriteMode

The readWriteMode definition is mainly used in the definition of filters to be applied in the time series display when used in edit mode. This element is an enumeration of;

- **read only** implies the data cannot be edited.
- **add originals** implies the data is new and is added to the database.
- **editing only visible to current task runs** implies any changes made remain invisible to other tasks (used in What-If scenarios)
- **editing visible to all future task runs** implies any changes made will be visible to other tasks
- **read originals only** implies all edited, corrected or interpolated data should be ignored.

The only enumeration that can be used in timeseriessets in FEWS modules is:

- **read complete forecast** reads the complete forecast series from the database. If this enumeration element is used, no Relative View Period has to be configured

It is a good convention to set this property to **read only** in all input blocks.

synchLevel

This is an integer value determining how the data is stored and synchronised through the distributed system or filtered when a database snapshot is created. There is no enumeration as the synchLevel is used in the configuration of synchronisation, where optimisations can be defined for each synchLevel. The convention used is explained in the Live System configuration section. Synclevel 0, 1, 2, 5, 6, 9 are automatically assigned when no synch level is configured depending on the the time series type and value type. see B Enumerations

expiryTime

This element allows the time series created to have a different expiry time to the default expiry time. This means it may be removed earlier, or later, by the rolling barrel function.

Attributes;

- **unit** (enumeration of: second, minute, hour, day, week)
- **multiplier** defines the number of units given above.
- **divider** same function as the multiplier, but defines fraction of units.

delay

This element allows the time series retrieved to be lagged (positive or negative). The time stamps of the series will then be shifted by the period specified on retrieval. This is used only when retrieving time series from the database, and not inversely when submitting time series to the database.

Attributes;

- **unit** (enumeration of: second, minute, hour, day, week)
- **multiplier** defines the number of units given above.
- **divider** same function as the multiplier, but defines fraction of units.

multiplier

This element allows the time series retrieved to be multiplied by the factor given. This is used only when retrieving time series from the database, and not inversely when submitting time series to the database.

divider

This element allows the time series retrieved to be divided by the factor given. This is used only when retrieving time series from the database, and not inversely when submitting time series to the database.

incrementer

This element allows the time series retrieved to be incremented by the factor given. This is used only when retrieving time series from the database, and not inversely when submitting time series to the database.

ensembleId

A time series set may be defined to retrieve all members of an ensemble at once (for example in evaluation of ensemble statistics). This is done by defining the optional ensembleId. The ensembleId should also be defined when writing new ensemble members. (e.g. on importing ensembles in the import module).

Example:

Example

```
            <timeSeriesSet>
        <moduleInstanceId>Import_NWS_GEFS</moduleInstanceId>
        <valueType>grid</valueType>
        <parameterId>P.forecast</parameterId>
        <locationId>GEFS</locationId>
        <timeSeriesType>external forecasting</timeSeriesType>
        <timeStep unit="hour" multiplier="6"/>
        <readWriteMode>add originals</readWriteMode>
        <ensembleId>GEFS</ensembleId>
    </timeSeriesSet>
```

When dealing with ensembles, the ensembleId needs only be defined if the workflow activity that is used must retrieve the complete ensemble, or if members are to be written under a new ensembleId. In other cases the ensembleId needs only be defined in the workflow definition (see Workflows chapter). For the TimeSeriesSets defined in modules there is then no difference between running in ensemble mode and running normally.

visibilityControllingFlagSourceColumnId

By defining a 'visibilityControllingFlagSourceColumnId' element in the timeSeriesSet, the existence of a flagSource Column for a particular timeseries (step) can be used as a condition for for example a transformation. When the configured flagSourceColumn is not existing for a particular timeseries step, the value will be perceived as 'Missing' for the transformation. All values that did not pass this validation step are removed on read. They are set to missing for equidistant timeseries and removed for non-equidistant timeseries. The flagSourceColumnId needs to be defined in flagSourceColumn.xml configuration file.

onlyReliableFlagSourceColumnId

By defining a 'onlyReliableFlagSourceColumnId' element in the timeSeriesSet, the existence of a flagSource Column for a particular timeseries (step) can be used as a condition for example a transformation or explorer filter, but then only for reliable data. When the configured flagSourceColumn is not existing for a particular timeseries step, the value will be perceived as 'Missing' for the transformation. All values that did not pass this validation step are removed on read. They are set to missing for equidistant timeseries and removed for non-equidistant timeseries. The flagSourceColumnId needs to be defined in flagSourceColumn.xml configuration file. This works the same as the above element visibilityControllingFlagSourceColumnId but then only for relaible data, it can be used together with a visibilityControllingFlagSourceColumnId, then the data needs to have either the visibilityControllingFlagSourceColumnId (the quality flag does not matter) or it need to have the onlyReliableFlagSourceColumnId and the data needs to be reliable.

qualifierAggregation

Can have values sum, mean, min or max.

When specified all the time series that have ALL (in the same time series set) specified qualifiers are aggregated to a single time series on read. Can not be specified when this time series set is meant for writing.

Key attributes

Key attributes are those attributes of a timeSeriesSet element, that distinguishes one timeSeriesSet from another. Key attributes are:

- moduleInstanceId / moduleInstanceSetId
- valuetype
- locationId/locationSetId
- parameterId
- domainParameterId (optional)
- timeSeriesType
- timeStep
- qualifierId (optional)
- ensembleId / ensembleMemberId / ensembleMemberIndexRange / ensembleMemberIndex (optional)
- aggregationPeriod (optional)
- cycle (optional)

Other attributes of the timeSerieSet only define what part of the whole timeserie Delft-FEWS should process (eg relativeViewPeriod), or what temporary transformation should be applied (eg incrementer).

Manual data edits

Timeseries can be edited by the user. This can be done from the explorer (filters.xml) or from the predefined displays (displaygroups.xml). It is required that the ReadWriteMode of the TimeSeriesSet is not set to "read only".There are some characteristics concerning handling of manually edited timeseries.

Automatic overwrite of manually edited values

- Manually corrected (not completed) imported values will not be rolled back when the value is reimported.
- Values added in the period after T0 in an external historical timeseries, WILL be overwritten by imported values.
- All other modules that generate the timeseries (such as a transformation) WILL overwrite manually edited data.

Expiry time of manually edited values

- A manually edited value will get the expiry time of the timeseriesset it is in, in the filters.xml or displaygroups.xml (depending on from which display the edit session started).
- If there is not expiryTime configured, then the default expirytime from the global.properties file will be taken.
- If this expiry time is not set, a default of 10 days is used.