

Import Module configuration options

| | |
|------------------------|---|
| What | <i>nameofinstance.xml</i> |
| Description | Configuration for import module |
| schema location | https://fewdocs.deltares.nl/schemas/version1.0/timeSeriesImportRun.xsd |

- Time Series Import Module
 - import
- general
 - description
 - importType
 - folder, ftp, sftp, server Url or jdbc connection
 - ftpPassiveMode
 - failedFolder
 - importTriggeringFile
 - groupImportPattern
 - fileNamePatternFilter
 - fileNameObservationDateTimePattern
 - fileNameForecastCreationDateTimePattern
 - fileNameEnsembleMemberIndexPattern
 - fileNameLocationIdPattern
 - fileNameParameterIdPattern
 - user
 - password
 - relativeViewPeriod
 - startDateTime
 - endDateTime
 - onlyGaps
 - externalForecastTimesSearchRelativePeriod & externalForecastTimesCardinalTimeStep
 - table
 - failOnUnmappableTimeSeries
 - failOnUnmappableLocations
 - logWarningsForUnmappableTimeSeries
 - logWarningsForUnmappableLocations
 - logWarningsForUnmappableParameters
 - logWarningsForUnmappableQualifiers
 - logWarningsToSeparateFile
 - ignoreFileNotFoundWarnings
 - idMapId
 - moduleInstanceAware
 - useStandardName (since stable build 2012.02)
 - maximumSnapDistance
 - maximumVerticalSnapDistance
 - unitConversionsId
 - flagConversionsId
 - geoDatum
 - missingValue
 - traceValue
 - importTimeZone
 - importTimeZone:timeZoneOffset
 - importTimeZone:timeZoneName
 - gridStartPoint
 - logErrorsAsWarnings and logErrorsAsWarningsToFileOnly
 - logMaxWarnings
 - dataFeedId
 - disableDataFeedInfo
 - overwriteAnnotations
 - trimPeriodWhenLastImportedTimeStepAfterPeriod
 - actionLogEventTypeId
 - comment
 - relativeStartTime
 - skipFirstLinesCount
 - validate
 - charset
- tolerance
- delay
- startTimeShift
 - startTimeShift:locationId
 - startTimeShift:parameterId
- temporary
- columnSeparator and decimalSeparator

- ignoreNonExistingLocationSets
- timeSeriesSet
- externUnit
- gridRecordTimeIgnore
- interpolateSerie
- Annotation location set id
- Example: Import of Meteosat images as time-series
- Packed files (zip, tar, tar.gz, tgz, gz, Z, tar.bz2, bz2, tbz2)
- EA Import module

Time Series Import Module

The time series import class can be applied to import data from a variety of external formats. The formats are included in an enumeration of supported import types. Each of these enumerations is used for a specifically formatted file.

fews:TimeSeriesImportRunComplexType

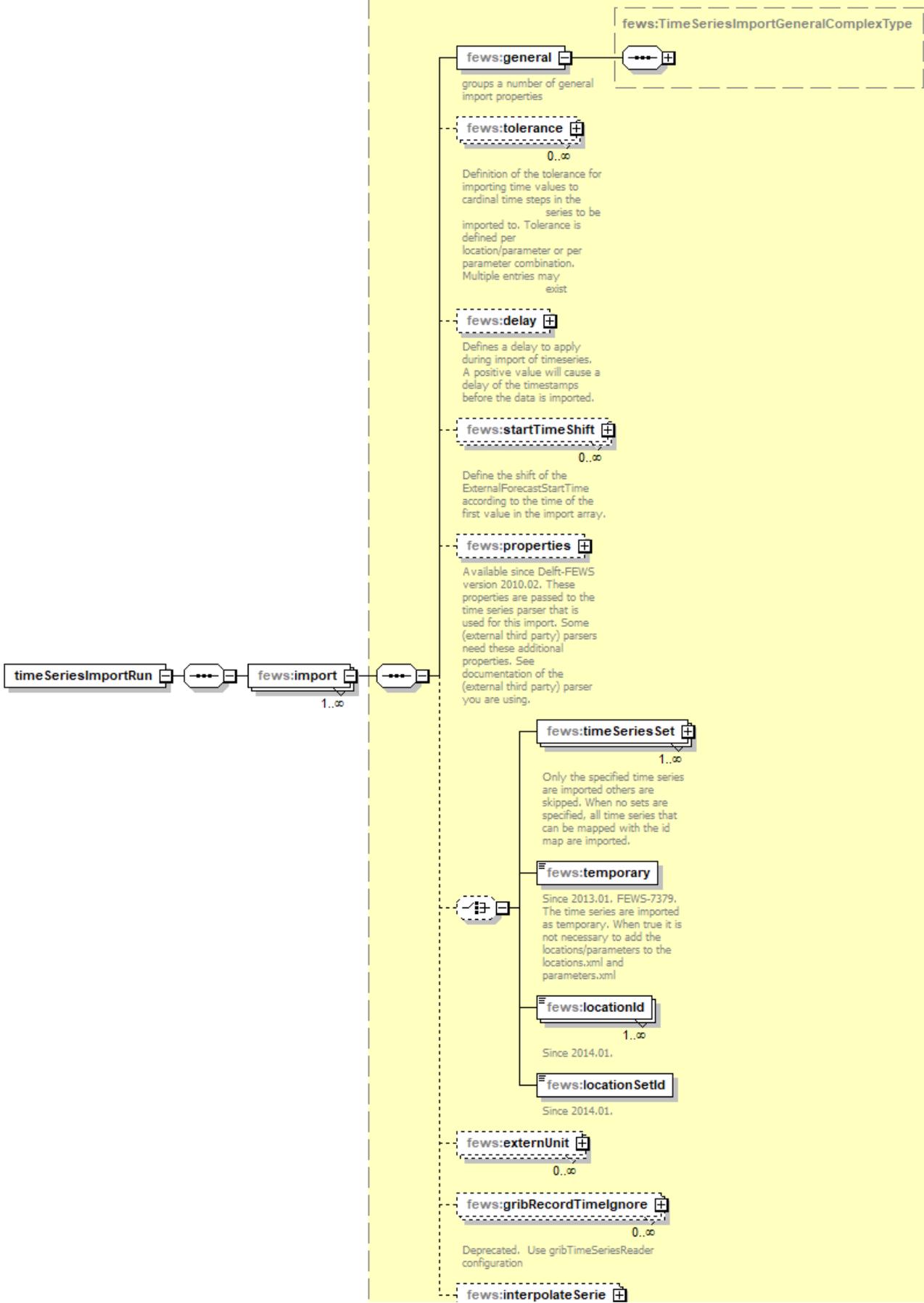




Figure 62 Elements of the TimeSeriesImport configuration

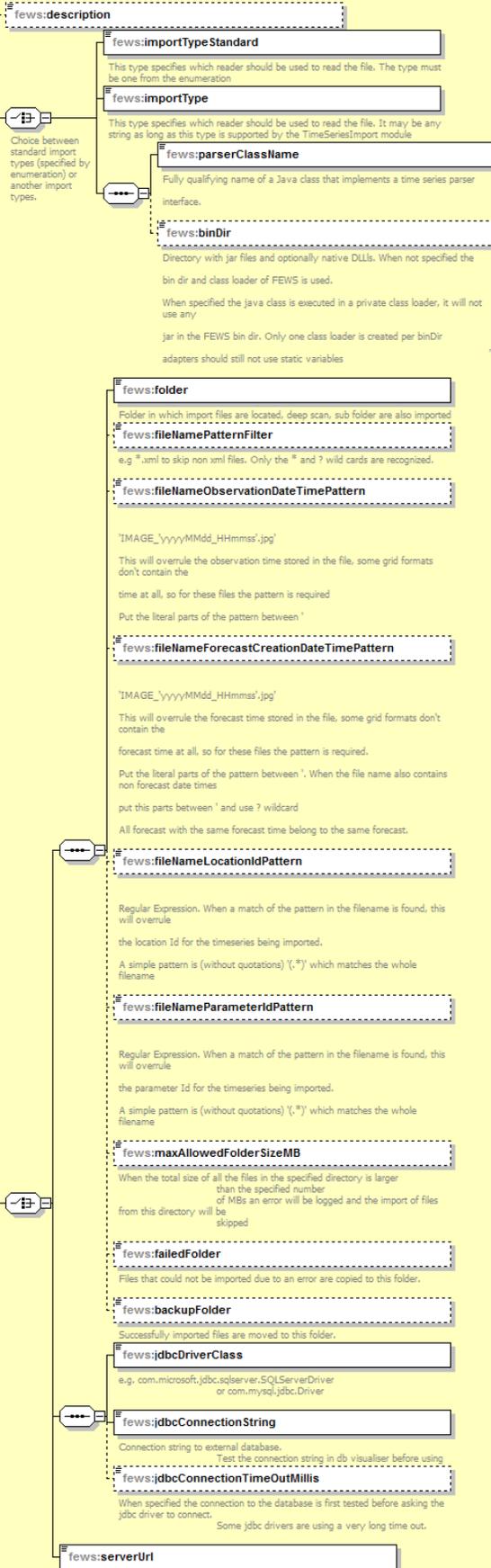
import

Root element for the definition of an import run task. Each task defined will import data in a specified format from a specified directory. For defining multiple formats, different import tasks reading from different directories must be defined.

general

Root element for general definitions used in the import runs.

fews:TimeSeriesImportGeneralComplexType



Choice between standard import types (specified by enumeration) or another import types.

Choice between standard import types (specified by enumeration) or another import types.

Choice between standard import types (specified by enumeration) or another import types.

Choice between standard import types (specified by enumeration) or another import types.

day or the time.
`%RELATIVE_TIME_IN_SECONDS(dateFormat:relativeTime)%` is replaced with `time0 + relativeTime`, where `time0` is the `timeZero` of this import run and `relativeTime` is a time relative to `time0` in seconds (can be negative). The time is formatted using the `dateFormat` that is specified as the first argument between the brackets. For example `%RELATIVE_TIME_IN_SECONDS(%yyyy/MMdd-18000)%` would be replaced with the year, month and day of time = `time0 - 18000` seconds.

fews:general

groups a number of general import properties

fews:UserAndPassword

User name and password, required for protected database connections and protected servers.

fews:user

User name used for authentication

fews:encryptedPassword

Encrypted user's password. Password is encrypted with clipboard utils in F12 menu in (any) SA/OC

fews:password

The user's password. Error will be logged when used. Error message will contain the encrypted password to replace the element with the element `encryptedPassword`

fews:relativeViewPeriod

Period relative to the time 0 of the run. When the start and end time are overulable the user can specify the download length with the cold state time and forecast length in the manual forecast dialog.

fews:startDateTime

Start date and time of the (fixed) period for which data should be read. Start is inclusive. This `dateTime` is in the configured `importTimeZone`.

fews:endDateTime

End date and time of the (fixed) period for which data should be read. End is inclusive. This `dateTime` is in the configured `importTimeZone`.

fews:forecastMaxAge

Skip all forecasts with are older than the specified age. The (external) forecast time and the time0 are used to calculate the age. The import file modification time and current actual time are not used.

fews:table

Currently the `generalCsv` and Database import require a table layout description configured by the user. Non-standard imports (plugins) can also require a table layout. See the documentation of the specific import.
 The order of the listed column is used to calculate the column indices when there is no column name information available. Add the columns that are not used as skipped columns so the column indices can be calculated correctly

fews:validate

Option to allow validation of the import files against the template, i.e. `xml-schema`. If there is no template available, this option will be ignored.

fews:logWarningsForUnmappableTime Series

When true warnings are logged when time series in the imported files are skipped. By default unmappable time series are silently skipped

fews:failOnUnmappableTime Series

If an import file contains some time series that can not be mapped use this option to mark file as failed

fews:logWarningsForUnmappableLocations

When true warnings are logged when locations in the imported files are skipped. By default unmappable locations are silently skipped

fews:failOnUnmappableLocations

If an import file contains some locations that can not be mapped use this option to mark file as failed

fews:idMapId

Id of IdMap to be used for parameter, location, qualifier and ensemble member mapping

fews:useStandardName

When the parser provides the standard name the parameter mapping can be done by the matching the standard name. The standard name of the parameters in the time series set should be configured in the `parameters.xml` and the standard name should be provided by the import format. If not an error is logged. When also the `maximumSnapDistance` is configured no id map is required at all

fews:maximumSnapDistance

In meters. When the parser provide location coordinates the location mapping can be by matching the coordinates. The snap distance is the tolerance used to detect the internal and external coordinates are the same. Don't forget to configure the geo datum when input format does not provide the coordinate system for the locations. When the parser does not provide the coordinates for a time series an error is logged

fews:unitConversionsId

Id of UnitConversions to be used for unit mapping

fews:disableImportOnMissingUnitConversion

If this flag is True and the extern unit does not equal the intern unit and there is no unit-mapping available, the time series will be not imported.

fews:flagConversionsId

Id of flagConversions to be used for flag mapping

fews:missingValue

Values that are replaced by NaN during import. NaN is always recognized as missing value.

fews:traceValue

All trace values are replaced by 0 during import, before datum conversion

fews:importTimeZone

Time zone of the import data. If this is not configured, then GMT is used.

fews:gridStartPoint

fews:geoDatum

Type of coordinate system

fews:importTypeConfig

configuration of the reader specified with importType

fevs:dataFeedId
Optional id for data feed. If not provided then the folder name will be used.

fevs:convertDatum
Convert datum from local datum during import. The conversion will be done for all parameters which use datum (as configured: Parameters.xml)

fevs:reportChangedValues
Option to allow printing messages to the log-file if the imported values are changed, compared to the previously imported values

fevs:actionLogEventTypeId
ID of the action message that must be logged if any data imported. This message is then used to start up an action as configured in the MasterController config files (e.g. start a forecast)

fevs:comment
Comment to add to the imported values, visible in the time series dialog as tooltip

fevs:synchLevel
Sync level used
when not specified in the time series sets

fevs:expiryTime
Expiry time used
when not specified in the time series sets

fevs:relativeForecastTime
Optional forecast time relative to the T0 of the import run. All imported external forecast time series will get this forecast time. This overrules any forecast times stored in the imported data itself. All time series with the same forecast time belong to the same forecast.

fevs:skipFirstLinesCount
Skips the first n lines of a file. Error is logged when this option is configured for a binary file.

description

Optional description for the import run. Used for reference purposes only.

importType

Specification of the format of the data to be imported. The enumeration of options includes for example:

- *MSW* : Import of data provided by the MSW System (Rijkswaterstaat, the Netherlands).
- *KNMI* : Import of synoptic data from KNMI (Dutch Meteorological Service).
- *WISKI* : Import of time series data from the WISKI Database system (Kisters AG).
- *DWD-GME* : Import of NWP data of the DWD Global Modell, (German Meteorological Service). This is a grid data format.
- *DWD-LM* : Import of NWP data of the DWD Lokal Modell, (German Meteorological Service). This is a grid data format.
- *GRIB* : Import of the GRIB data format. General format for exchange of meteorological data.
- *EVN*: Import of data in the EVN format (Austrian Telemetry)
- *METEOSAT*: Import of images from meteosat satellite

The complete list of possible importTypes is listed [here](#)

folder, ftp, sftp, server Url or jdbc connection

Location to import data from. This may be a UNC path (ie located on the network), sftp, http or from a database.

JDBC example:

```

<general>
  <importTypeStandard>database</importTypeStandard>
  <jdbcDriverClass>com.mysql.jdbc.Driver</jdbcDriverClass>
  <jdbcConnectionString>jdbc:mysql://192.168.101.215/cwb_ac</jdbcConnectionString>
  <user>sobek</user>
  <password>Tohek<w>cwa</password>
  <relativeViewPeriod startOverrutable="true" endOverrutable="true" start="-1" end="1" unit="day"/>
  <table name="qpe_sums_obs">
    <dateTimeColumn name="rehdate"/>
    <valueColumn name="rad_gz" unit="mm/hr" locationId="Qesums" parameterId="P.radar.actual" parser="Mosaic"
  />
  </table>
  <table name="qpe_sums_fo">
    <forecastDateTimeColumn name="createdate"/>
    <dateTimeColumn name="raddate"/>
    <valueColumn name="rad_gz" unit="mm/hr" locationId="Qpesums" parameterId="P.radar.forecast" parser="
Mosaic"/>
  </table>
  <unitConversionsId>ImportUnitConversions</unitConversionsId>
  <importTimeZone>
    <timeZoneOffset>+00:00</timeZoneOffset>
  </importTimeZone>
  <dataFeedId>QPE_Sum</dataFeedId>
</general>

```

sftp example:

```

<general>
  <importType>TypicalAsciiForecast</importType>
  <folder>sftp://wf:2004wrf@130.167.66.114:22/home/WRF</folder>
  <relativeViewPeriod startOverrutable="true" endOverrutable="true" start="-1" end="3" unit="day"/>
  <unitConversionsId>ImportUnitConversions</unitConversionsId>
  <importTimeZone>
    <timeZoneOffset>+00:00</timeZoneOffset>
  </importTimeZone>
  <dataFeedId>Forecast</dataFeedId>
</general>

```

http example:

```

<general>
  <importType>RemoteServer</importType>
  <serverUrl>http://192.168.65.12:8002</serverUrl>
  <relativeViewPeriod startOverrutable="true" endOverrutable="true" start="-1" end="0" unit="day"/>
  <idMapId>IdImportRO</idMapId>
  <importTimeZone>
    <timeZoneOffset>+10:00</timeZoneOffset>
  </importTimeZone>
  <dataFeedId>RO</dataFeedId>
</general>

```

When using the serverUrl (http), then it is also possible to use tags in the serverUrl. Tags should be separated by "%" signs. The following tags can be used in this URL. %TIME_ZERO(dateFormat)% is replaced with the time zero of the import run. The time zero is formatted using the dateFormat that is specified between the brackets. For example %TIME_ZERO(yyyyMMdd)% would be replaced with the year, month and day of the time zero. %RELATIVE_TIME_IN_SECONDS(dateFormat,relativeTime)% is replaced with time = (time0 + relativeTime), where time0 is the time zero of the import run and relativeTime is a time relative to time0 in seconds (can be negative). The time is formatted using the dateFormat that is specified as the first argument between the brackets. For example %RELATIVE_TIME_IN_SECONDS(yyyyMMdd,-18000)% would be replaced with the year, month and day of time = time0 - 18000 seconds. Examples of serverUrls with tags in it:

```

<serverUrl>http://nomads.ncep.noaa.gov:9090/dods/gfs/gfs%TIME_ZERO(yyyyMMdd)%/gfs_%TIME_ZERO(HH)%z</serverUrl>

```

```

<serverUrl>http://nomads.ncep.noaa.gov:9090/dods/gfs/gfs%RELATIVE_TIME_IN_SECONDS(yyyyMMdd, -18000 )%/gfs_%
RELATIVE_TIME_IN_SECONDS(HH,-18000)%z</serverUrl>

```

ftpPassiveMode

When using "ftp://" import the option of passive mode is supported. This is an optional field, default value is false, and can be configured in the general section.

```
<general>
    <importType>importType</importType>
    <folder>ftp://import/external</folder>
    <ftpPassiveMode>true</ftpPassiveMode>
</general>
```

failedFolder

Folder to move badly formatted files to. This may be a UNC path (ie located on the network).

importTriggeringFile

Use this option to configure a path to one or more files that trigger the import. The import will only start when all triggering files are available. The triggering files will be deleted at the end of the import run.

Some examples:

```
<importTriggeringFile>p:\myProject\trigger</importTriggeringFile>
```

or

```
<importTriggeringFile>$REGION_HOME$/import/external/trigger1</importTriggeringFile>
<importTriggeringFile>$REGION_HOME$/import/external/trigger2</importTriggeringFile>
```

Please note that the triggering files cannot be located on the FTP.

groupImportPattern

Since 2024.01 - Option to specify groups of files that need to be imported together. This can be useful for forecasts spread over several files which should be imported as one forecast i.e. a group of files, in particular if each file is added to the import folder at a different time.

The filenames must contain the forecast date and time. The pattern to define a group is specified in `fileNameDateTimePattern`, with the expected number of files for one forecast in `numberOfFiles`. The files will be imported only when all expected files are in the import folder. A time span also has to be specified in `waitingTime` to define when to consider the forecast as failed and move it to the failed folder. If more than the waiting time has passed between the forecast time and T0, the forecast is considered failed and the files are moved to the failed folder.

For example, for the following configuration:

```
<groupImportPattern>
    <numberOfFiles>6</numberOfFiles>
    <fileNameDateTimePattern>'timeseries_'MMddHHmm'_'?</fileNameDateTimePattern>
    <waitingTime unit="day" multiplier="1"/>
</groupImportPattern>
```

with the import folder containing the files:

- timeseries_01011200_1
- timeseries_01011200_2
- timeseries_01011200_3
- timeseries_01011200_4
- timeseries_01011200_5
- timeseries_01011200_6
- timeseries_01021200_1

The first six files will be imported because six files are matching the pattern "timeseries_01011200_?". But the last file will not be imported yet because it is the only file matching the pattern "timeseries_01021200_?" .

fileNamePatternFilter

Option to filter files that need to be imported based on their filename. This can be useful if there are files in the import folder that should not be imported:

e.g *.xml to skip non xml files.

Another usecase is when you are importing a large number of big files which FEWS cannot import all at once due to memory issues (eg reanalysis data). In this case you can use the T0 of the system to filter out files you want to import in one run:

eg: <fileNamePatternFilter>%TIME_ZERO(yyyy)%?????.nc</fileNamePatternFilter>

fileNameObservationDateTimePattern

'IMAGE_'yyyyMMdd_HHmss'.jpg' This will overrule the observation time stored in the file, some grid formats don't contain the time at all, so for these files the pattern is required. Put the literal parts of the pattern between single quotes. If the literal parts contain unrelated numbers or an unrelated date, replace each digit with a ?. The ? should still be between the single quotes.

Example: T20190524_20190722 'T?????????'_yyyyMMdd.

Currently supported only by import from OpenDAP.

fileNameForecastCreationDateTimePattern

'IMAGE_'yyyyMMdd_HHmss'.jpg' This will overrule the forecast time stored in the file, some grid formats don't contain the forecast time at all, so for these files the pattern is required. Put the literal parts of the pattern between single quotes. When the file name also contains non forecast date times. Put this parts between ' and use ? wildcard. All forecast with the same forecast time belong to the same forecast, the "fileNameObservationDateTimePattern" is not required in that case.

Example: MSPM_<observation date time>_<forecast date time>.asc 'MSPM_?????????????????'_yyyyMMddHHmss'.asc'

When filename starts with the pattern do not use quotes in at the start: yyyyMMdd'_bla.nc'

fileNameEnsembleMemberIndexPattern

Use ? to indicate the position of the ensemble member index in the filename. For example

<fileNameEnsembleMemberIndexPattern>cosmo_???.dat</fileNameEnsembleMemberIndexPattern>

Litteral parts that contain unrelated characters can be also replaced with a dummy character. For example

<fileNameEnsembleMemberIndexPattern>xxxxx???.dat</fileNameEnsembleMemberIndexPattern>

This is useful if the filename keeps changing , for example because of the presence of a date in the filename

fileNameLocationIdPattern

Use the filename, or part of it, as the locationId for the TimeSeries that will be imported, using Regular Expressions. When a match of the pattern in the filename is found, this will overrule the location Id for the timeseries being imported.

examples

| file name | pattern | location id |
|----------------------------------|----------------|-------------|
| PObs-1234.txt | [^-][-](.*){4} | 1234 |
| H1234.txt | .{1}(.*){4} | 1234 |
| BAFU-2021-PegelRadarSchacht.csv | (.*)[-][^]* | BAFU-2021 |
| Pegel-Radar-Schacht-BAFU2021.csv | ([^*]*)\.csv | BAFU2021 |

You can test and build your expression with

<https://regex101.com>

The location id is the "group 1" in the match information. Only one output is allowed.

fileNameParameterIdPattern

Use the filename, or part of it, as the parameterId for the TimeSeries that will be imported, using Regular Expressions. See description above for fileNameLocationIdPattern.

user

User name, required when importing from protected database connections or protected servers.

password

Password, required when importing from protected database connections or protected servers. Please use the Hex value for special characters (e.g. a @ must be specified as %40)

relativeViewPeriod

The relative period for which data should be imported. This period is relative to the time 0 of the run. When the start and end time are overrullable the user can specify the download length with the cold state time and forecast length in the manual forecast dialog. It is also possible to import data for an absolute period of time using the startDateTime and endDateTime elements.

startDateTime

Start date and time of the (absolute) period for which data should be imported. Start is inclusive. This dateTime is in the configured importTimeZone. It is also possible to import data for a relative period of time using the relativeViewPeriod element.

endDateTime

End date and time of the (absolute) period for which data should be imported. End is inclusive. This dateTime is in the configured importTimeZone. It is also possible to import data for a relative period of time using the relativeViewPeriod element.

onlyGaps

Since 2022.02. Specifically designed for service imports.

This element specifies that only for periods with gaps (missing data enclosed within non missing data) in the import time series data will be imported.

First all gaps of the import time series within the period for the import are detected, then the import is run for all periods of the gaps.

A gap in any time series will result in the (re)import for all time series.

Non equidistant series will not be included for detection of gaps.

Unreliable non-missing values are not counted as gaps.

Missing values at the start and end of a configured period for a series are not counted as gaps.

This element needs to be combined with either a relativeViewPeriod or a startDateTime and endDateTime in the general part of the import config.

externalForecastTimesSearchRelativePeriod & externalForecastTimesCardinalTimeStep

For each externalForecastTimesCardinalTimeStep within externalForecastTimesSearchRelativePeriod, multiple external forecasts are requested to be imported. This way it is possible to import data for multiple (older) forecast times and not just the most recent. Only supported by specific import types, like [WIWB](#).

table

Currently the generalCsv and Database import require a table layout description configured by the user. Non-standard imports (plugins) can also require a table layout. See the documentation of the specific import.

[Table Layout](#)

failOnUnmappableTimeSeries

This optional element lets the imported file to be moved to the failedFolder in case the file contains time series that are not mapped to time series in the import module. Very usefull for testing if you do not expect unknown time series to be imported. Default value is FALSE.

failOnUnmappableLocations

This optional element lets the imported file to be moved to the failedFolder in case the file contains locations that are not mapped to locations in the import module. Very usefull for testing if you do not expect unknown time series to be imported. Default value is FALSE.

logWarningsForUnmappableTimeSeries

When true warnings are logged when time series in the imported files are skipped. By default unmappable time series are silently skipped

logWarningsForUnmappableLocations

When true warnings are logged when locations in the imported files are skipped. By default unmappable locations are silently skipped

logWarningsForUnmappableParameters

When true warnings are logged when parameters in the imported files are skipped. By default unmappable parameters are silently skipped

logWarningsForUnmappableQualifiers

When true warnings are logged when qualifiers in the imported files are skipped. By default unmappable qualifiers are silently skipped

logWarningsToSeparateFile

Log warnings for import file to a specific file called [importFileName].log. This file will be placed in either the backupFolder or failedFolder depending on whether they are configured and if the import was succesful or not. This setting does not work for OpenDap imports

ignoreFileNotFoundWarnings

Available since 2020.02. Default is false. If it is set to true, FileNotFound warnings are only logged in debug mode. This is a useful options in cases where a fixed list of files is sourced (eg one file per location in a locationSet), but where incidentally files are not available.

idMapId

ID of the IdMap used to convert external parameterId's and locationId's to internal parameter and location Id's. Each of the formats specified will have a unique method of identifying the id in the external format. See section on configuration for Mapping Id's units and flags.

moduleInstanceAware

Since 2023.01. If value is set to true, data will only be imported if the module instance id specified in the config file matches the module instance id of the downloaded data.

useStandardName (since stable build 2012.02)

Optional. When the parser provides the standard name the parameter mapping can be done by matching the standard name. The standard name of the parameter in the time series set should be configured in the parameters.xml and the standard name should be provided by the import format. If not an error is logged. When also the maximumSnapDistance is configured no id map is required at all.

maximumSnapDistance

Since 2012.02. Optional maximum horizontal snap distance in meters. When the parser provides horizontal location coordinates (x,y) and no locationIds, then the location mapping will be done by matching the horizontal coordinates. The horizontal snap distance is the tolerance used to detect which internal and external horizontal coordinates are the same. Don't forget to configure the geoDatum when the input format does not provide the coordinate system for the locations. When the parser does not provide the coordinates for a time series an error is logged. Note: this option has no effect for grid data. Note 2: it is not possible to import data using horizontal coordinates and using locationIds in the same import, need to define separate import elements for that (one with maximumSnapDistance and one without maximumSnapDistance).

maximumVerticalSnapDistance

Since 2014.02. Optional maximum vertical snap distance in meters. When the parser provides vertical location coordinates (z) and no locationIds, then the location mapping will be done by matching the vertical coordinates. The vertical snap distance is the tolerance used to detect which internal and external vertical coordinates are the same. This only works when the input format provides the coordinates of the locations. When the parser does not provide the vertical coordinates for a time series an error is logged. Note: this option currently only works for importing horizontal layers from netcdf 3D grid data. Note 2: it is not possible to import data with z-coordinates (layers from 3D grids) and data without z-coordinates (2D grids) in the same import, need to define separate import elements for that (one with maximumVerticalSnapDistance and one without maximumVerticalSnapDistance).

unitConversionsId

ID of the UnitConversions used to convert external units to internal units. Each of the formats specified will have a unique method of identifying the unit in the external format. See section on configuration for Mapping Id's units and flags.

flagConversionsId

ID of the FlagConversions used to convert external data quality flags to internal data quality flags. Each of the formats specified will have a unique method of identifying the flag in the external format. See section on configuration for Mapping Id's units and flags.

geoDatum

Convert the geographical coordinate system (horizontal datum and projection) to specified geoDatum during import. Not all parsers support this parameter so please check the documentation for a particular parser to see if it is supported.

missingValue

Optional specification of missing value identifier in external data format.

traceValue

Value to be replaced by 0 during import, before datum conversion. For example, some NWP products show minute negative values. These can be filtered upon import by setting the traceValue. Example: <traceValue>-0.025</traceValue> would replace all instances of -0.025 by 0.

importTimeZone

Time zone the external data is provided in if this is not specified in the data format itself. The timezone in the data format will always override the timezone configured using this option. This may be specified as a timeZoneOffset, or as a specific timeZoneName.

importTimeZone:timeZoneOffset

The offset of the time zone with reference to UTC (equivalent to GMT). Entries should define the number of hours (or fraction of hours) offset. (e.g. +01:00)

importTimeZone:timeZoneName

Enumeration of supported time zones. See appendix B for list of supported time zones.

gridStartPoint

Identification of the cell considered as the first cell of the grid. This option should only be used if the NetCDF file is not CF compliant and contains insufficient metadata with info of the grid, its orientation etc.

Enumeration of options include :

- *NW* : for upper left
- *SW* : for lower left
- *NE* : for upper right
- *SE* : for lower right

gridStartPoint is supported by the import type netcdf-cf_grid, matroos_netcdfmapseries , grib1, grib2 and cemig
Options NE and SE are supported only by NETCDF-CF_GRID

logErrorsAsWarnings and logErrorsAsWarningsToFileOnly

These two options are a choice. Default is logErrorsAsWarnings=true, logErrorsAsWarningsToFileOnly = false. Exceptions occurring in a parser as well as some not parser-specific log messages such as " Import folder ... does not exist" or " Can not connect to.." can be logged as Error or as Warning. Use this option to change it. Configure logErrorsAsWarnings=false if you wish clear alert notifications in SystemMonitor and Explorer statusBar. Configure logErrorsAsWarningsToFileOnly=true if you do not want the warnings to be saved in the database.

logErrorsAsWarnings available since 2014.01
logErrorsAsWarningsToFileOnly available since 2018.02

logMaxWarnings

Maximum number of warnings logged for import. The default is 5. This setting does not work for OpenDap imports.

dataFeedId

Optional id for data feed. If not provided then the folder name will be used. This is used in the [SystemMonitorDisplay in the importstatus tab](#).

disableDataFeedInfo

By default all data feeds are visible in [SystemMonitorDisplay in the importstatus tab](#). If some data feeds are not wanted, for example because they are not really relevant, one can use this option

overwriteAnnotations

Configuring `<overwriteAnnotations>true</overwriteAnnotations>` will make sure that all existing annotations with the same location - time combination will be overwritten when a new annotation is imported. By default annotations with different content in either the annotation itself or one of the configurable properties will exist next to eachother.

trimPeriodWhenLastImportedTimeStepAfterPeriod

Since 2018.02. When true, import types that trim requested period to last imported time step (like [WIWB](#)) will not make an exception (and skip entire period) when last imported time step is after the requested period

actionLogEventTypeId

ID of the action message that must be logged if any data imported. This message is then used to start up an action as configured in the MasterController config.files (e.g. start a forecast)

comment

You can add a comment by the importmodule to the first imported value (or all values) by using the commentForFirstValue or commentForAllValues element.

The next tags are possible within the comment: %IMPORT_DATE_TIME%, %FILE_DATE_TIME%, %FILE_NAME%
Time formats can be configured through the timeZone and dateTimePattern elements.

relativeStartTime

Optional forecast time relative to the T0 of the import run. All imported external forecast time series will get this forecast time. This overrules any forecast times stored in the imported data itself. All time series with the same forecast time belong to the same forecast.

skipFirstLinesCount

Skips the first n lines of a ASCII file (like CSV). Error is logged when this option is configured for a binary file.

validate

Option to allow validation of the import files against the template, i.e. xml-schema. If there is no template available, this option will be ignored.

charset

Since 2017.01 it is possible to explicitly configure the charset for text file imports, like the generalCsv import.

tolerance

Definition of the tolerance for importing time values to cardinal time steps in the series to be imported to. Tolerance is defined per location/parameter combination. Multiple entries may exist.

Attributes;

- *locationId* : Id of the location tolerance is to be considered for.
- *locationSetId* : Id of the location set tolerance is to be considered for.
- *parameterId* : Id for the parameter tolerance is to be considered for.
- *timeUnit* : Specification of time units tolerances is defined in (enumeration).
- *unitCount* : integer number of units defined for tolerance.

delay

Specification of a delay to apply to all the time stamps of imported time series. The delay is defined by a time unit and a multiplier. A negative delay shifts the time series back in time.

Attributes;

- *locationId* : Id of the location for which to apply the delay.
- *locationSetId* : Id of the location for which to apply the delay.
- *parameterId* : Id for the parameter for which to apply the delay.
- *timeUnit* : Specification of time units the delay is defined in.
- *multiplier* : Integer number of time units for the delay.

For example, to specify a delay of 2 hours, enter "hour" as unit and 2 as multiplier.

Note: You cannot use this option for timeseries of type external forecasting because doing so would require that the same delay is also applied to the forecast time.

startTimeShift

Specification of a shift to apply to the start time of a data series to be imported as external forecasting. This is required when the time value of the first data point is not the same as the start time of the forecast. This may be the case in for example external precipitation values, where the first value given is the accumulative precipitation for the first time step. The start time of the forecast is then one time unit earlier than the first data point in the series. Multiple entries may exist.

`startTimeShift:locationId`

Id of the location to apply the startTimeShift to.

`startTimeShift:parameterId`

Id of the parameter to apply the startTimeShift to.

temporary

Since 2013.01. The time series are imported as temporary. When true it is not necessary to add the locations/parameters to the locations.xml and parameters.xml

columnSeparator and decimalSeparator

Since 2016.01 (so far only implemented for GeneralCsv import type) it is possible to choose from multiple column separators: comma ",", or semi-colon ";", or pipe "|" or tab "	" or space " "

When specifying a column separator it is compulsory to also specify the decimal separator as comma ",", or point "."

For example see [generalCsv](#) import type.

ignoreNonExistingLocationSets

Since 2021.01 it is possible to set ignoreNonExistingLocationSets to true. Then time series sets which contain a non existing location set id will be ignored instead of resulting in an error. This is very useful for combining template loops where multiple tags are replaced and some combinations of tags are invalid.

properties

Available since Delft-FEWS version 2010.02. These properties are passed to the time series parser that is used for this import. Some (external third party) parsers need these additional properties. See documentation of the (external third party) parser you are using.

timeSeriesSet

TimeSeriesSet to import the data to. Multiple time series sets may be defined, and each may include either a (list of) locationId's or a locationSetId. Data imported is first read from the source data file in the format specified. An attempt is then made to map the locationId's and the parameterId's as specified in the IdMap's to one of the locations/parameters defined in the import time series sets. If a valid match is found, then the time values are mapped to those in the timeSeriesSet, taking into account the tolerance for time values. A new entry is made in the timeSeries for each valid match made.

For non-equidistant time series the time values imported will be taken as is. For equidistant time series values are only returned on the cardinal time steps. For cardinal time steps where no value is available, no data is returned.

externUnit

For some data formats an external unit is not defined in the file to be imported. The element <externUnit> allows the unit to be specified explicitly, which is then used to find the corresponding unit conversions as configured in a UnitConversionsFiles (see [02 Unit Conversions](#)).

```
<externUnit parameterId="P.nwp.fcst" unit="mm" cumulativeSum="false"/>
```

Attributes of <externUnit>:

- parameterId: Id of the parameter for which a unit is specified. This is the internal parameter Id.
- unit: specification of unit. This unit must be available in the UnitConversions specified in the unitConversionsId element.
- cumulativeSum: if this option is set to "true", then it is possible to receive the parameters as cumulative sums. Cumulative sums are partial sums of a given sequence of numbers. For example, if the sequence is: {a, b, c, d, ...}, then the cumulative sums are: a, a+b, a+b+c, a+b+c+d, The import module will then calculate the values for the individual time steps {a, b, c, d, ...} by subtracting the cumulative sum of the previous cumulative sum value.
- cumulativeMean: similar to *cumulativeSum*, if cumulativeMean is set to "true", then it is possible to receive the parameters as cumulative means. Cumulative means are partial means of a given sequence of numbers. Therefore, for a given sequence of numbers $s = \{x_1, x_2, \dots, x_n, x_{n+1}\}$ cumulative means are calculated as follows: $CM(x_{n+1}) = [x_{n+1} + n * CM(x_n)] / (n+1)$. The import module will then calculate the values for the individual time steps {a, b, c, d, ...} by multiplying the cumulative mean with the amount of time steps already processed and then subtracting of the previous value that has been also multiplied with the amount of timestep processed minus 1.

gridRecordTimeIgnore

Boolean flag to specify if the start of forecast is read from the GRIB file or if it is inferred from the data imported. In some GRIB files a start of forecast is specified, but the definition of this may differ from that used in DELFT-FEWS.



When importing grid data from file formats where the attributes of the grid is not specified in the file being imported (ie the file is not self-describing), a definition of the grid should be included in the Grids configuration (see Regional Configuration).

It is also advisable to define the Grid attributes for self-describing Grids such as those imported from GRIB files. If no GRIB data is available, then DELFT-FEWS will require a specification of the grid to allow a Missing values grid to be created.

interpolateSerie

The timestep of some datasets grows larger as their forecast goes further into the future. For example; the regular timestep is 3 hours, but after 24 hours it increases to 6 hours. This element then allows the dataset to be imported with a 3 hour timestep and will interpolate when the dataset reaches 6 hours.

Attributes;

- parameterId: Id of the parameter for which a unit is specified. This is the internal parameter Id.
- interpolate: boolean flag to specify whether to interpolate for this parameter or not.

Annotation location set id

Since 2021.01 it is possible to import annotations via the generalCsv type.

It will import all annotations that exist for the configured <annotationLocationSetId>

The value column will be used for the text of the annotation itself.

Example annotations import

```
<?xml version="1.0" encoding="UTF-8"?>
<!--Delft FEWS Maas -->
<timeSeriesImportRun 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
/timeSeriesImportRun.xsd">
  <import>
    <general>
      <importType>generalCSV</importType>
      <folder>import/annotations</folder>
      <table>
        <dateTimeColumn name="CREATION_TIME" pattern="dd-MM-yyyy HH:mm"/>
        <startDateTimeColumn name="START" pattern="dd-MM-yyyy HH:mm"/>
        <endDateTimeColumn name="END" pattern="dd-MM-yyyy HH:mm"/>
        <locationColumn name="LOC"/>
        <valueColumn name="ANNOTATION"/>
        <propertyColumn name="CATEGORIE" key="Categorie"/>
        <propertyColumn name="SUB-CATEGORIE" key="Sub-Categorie"/>
      </table>
    </general>
    <annotationLocationSetId>Stand alone Stations</annotationLocationSetId>
  </import>
</timeSeriesImportRun>
```

There are special rules when it comes to overwriting annotations for the same location-time combination.

When annotations have different content for either the annotation text (valueColumn) or any of the properties (propertyColumn), they are considered unique annotations and will exist next to each other (this is solved in the background of FEWS by shifting newer annotations by milliseconds).

When an annotation is imported with the exact same location-time combination and identical contents of the annotation text (valueColumn) and all properties (propertyColumn) for an already existing annotation, then it is checked whether the start and or end time has changed and if so, they are updated. But when the start and end time are also the same the "new" annotation is considered the same and any potential changes in other columns are ignored and the annotation will not be imported.

This behaviour can be overruled with the <overwriteAnnotations>true</overwriteAnnotations> which will make sure that all existing annotations with the same location - time combination will be overwritten when a new annotation is imported.

Example: Import of Meteosat images as time-series

Meteosat Images are generally imported as images in *[filename].png* format. The Meteosat images constitute a time series of png images, that are geo-referenced by means of a specific world file. Each image needs its own world file, which in case of PNG carries the extension *[filename].pgw*. Import of images in another format, such as JPEG is also possible. The corresponding world file for a JPEG file has the extension *[filename].jpg*. The images are imported via a common time series import, for which a specific image parameter needs to be specified in a *parameterGroup* via the parameter *id image*.

```
<parameterGroup id="image">
  <parameterType>instantaneous</parameterType>
  <unit>-</unit>
  <valueResolution>8</valueResolution>
  <parameter id="image">
    <shortName>image</shortName>
  </parameter>
</parameterGroup>
```

The value resolution indicates the resolution of the values of the pixels (grey tones) in the Meteosat images. In this case 8 grey tones are resampled into a single grey tone for storage space reductions. In the module for the timeseries import run for a Meteosat image the import is then configured as follows:

```
<import>
  <general>
    <importType>GrayScaleImage</importType>
    <folder>${REGIONHOME}/Import/MeteoSat</folder>
    <idMapId>IdImportMeteosat</idMapId>
  </general>

  <timeSeriesSet>
    <moduleInstanceId>ImportMeteosat</moduleInstanceId>
    <valueType>grid</valueType>
    <parameterId>image</parameterId>
    <locationId>meteosat</locationId>
    <timeSeriesType>external historical</timeSeriesType>
    <timeStep unit="minute" multiplier="15"/>
    <readWriteMode>add originals</readWriteMode>
    <synchLevel>4</synchLevel>
    <expiryTime unit="day" multiplier="750"/>
  </timeSeriesSet>
</import>
```

The geo-referenced image can then be displayed in the grid display.

Packed files (zip, tar, tar.gz, tgz, gz, Z, tar.bz2, bz2, tbz2)

Packed files are imported as a folder. The directory structure in the file is ignored, all levels are imported. The files are unpacked while reading the file. This happens in chunks so it will not require extra memory. Files that require random access, like NetCDF files, are unpacked to a temporary file to the local file system.

The following packed file extensions are supported

- zip
- tar
- tar.gz
- tgz
- gz
- Z
- tar.bz2
- bz2
- tbz2

EA Import module

A specific import class is available for importing time series data from the XML format specified by the UK Environment Agency. The configuration items required are a sub-set of those required in the more generic time series import format. This is due to much of the required information being available in the XML file itself (ie file is self describing).

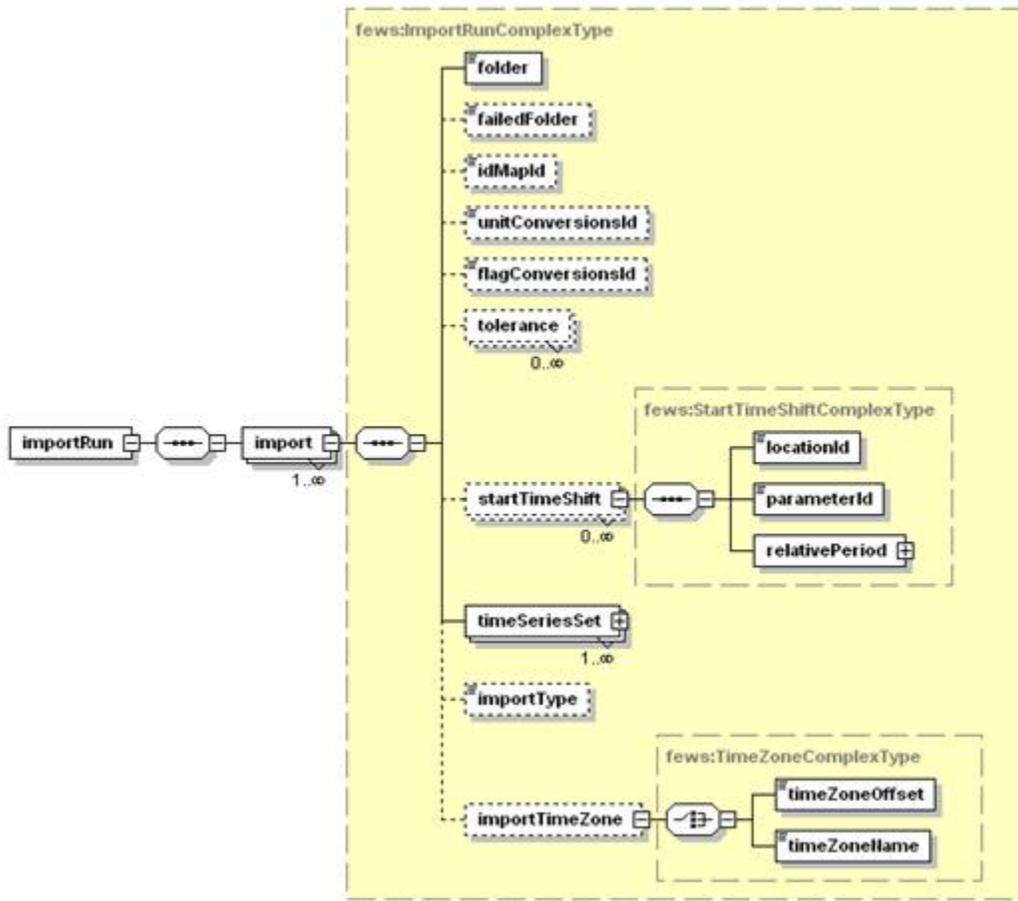


Figure 63 Elements of the EAImport configuration.