Deltares

enabling delta life

Deltares

enabling delta life

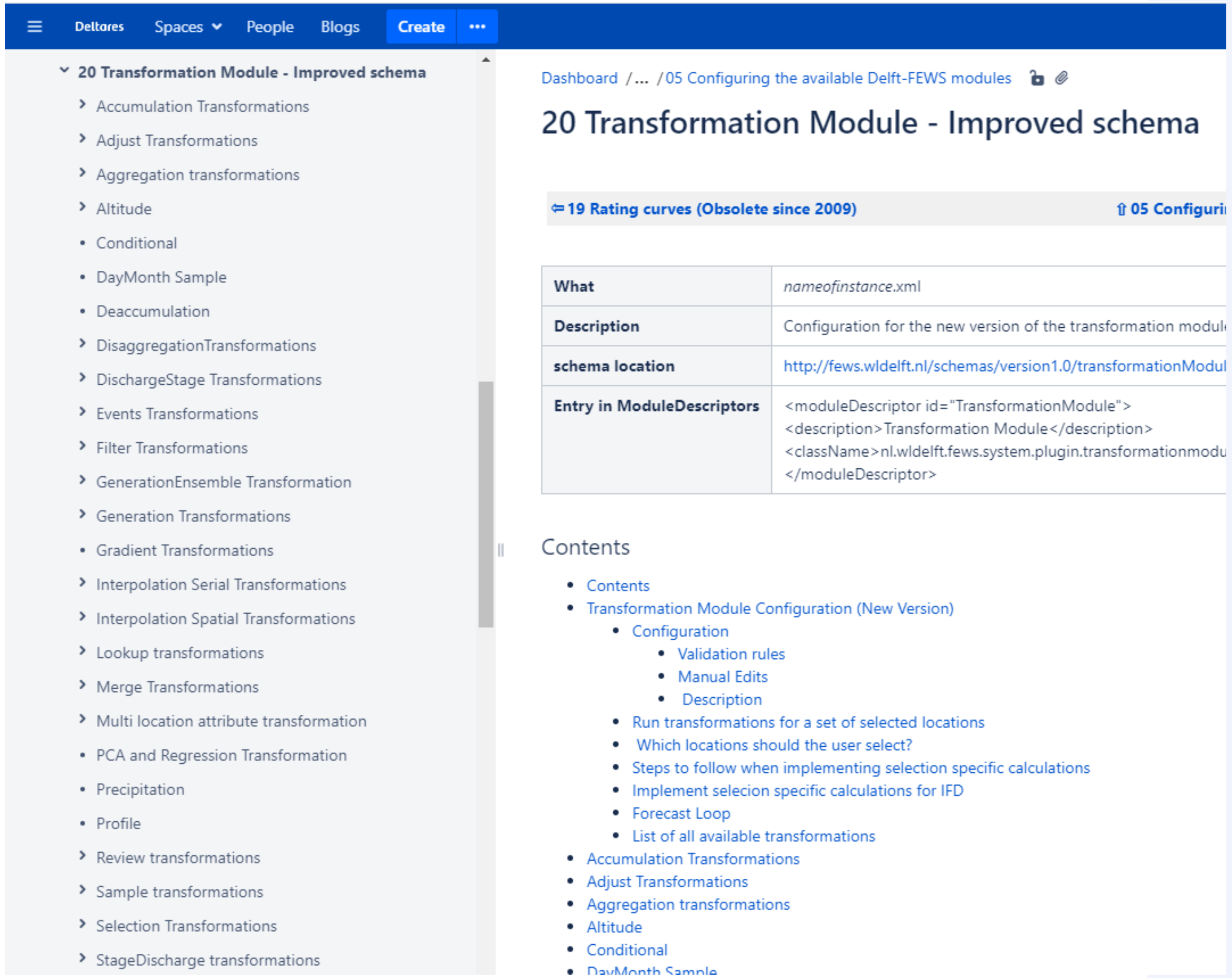
Delft-FEWS

Basic Configuration Course

Module: Processing Data with Transformations

Module Motivation

- Data direct from source is not often ready for direct use.
- It needs to be processed for quality control, for visualization and for use in models.
- Transformations in Delft-FEWS can be incredibly powerful, useful and not difficult to implement.



- 20 Transformation Module - Improved schema
  - Accumulation Transformations
  - Adjust Transformations
  - Aggregation transformations
  - Altitude
  - Conditional
  - DayMonth Sample
  - Deaccumulation
  - DisaggregationTransformations
  - DischargeStage Transformations
  - Events Transformations
  - Filter Transformations
  - GenerationEnsemble Transformation
  - Generation Transformations
  - Gradient Transformations
  - Interpolation Serial Transformations
  - Interpolation Spatial Transformations
  - Lookup transformations
  - Merge Transformations
  - Multi location attribute transformation
  - PCA and Regression Transformation
  - Precipitation
  - Profile
  - Review transformations
  - Sample transformations
  - Selection Transformations
  - StageDischarge transformations

Dashboard /... /05 Configuring the available Delft-FEWS modules

20 Transformation Module - Improved schema

⇦19 Rating curves (Obsolete since 2009) ⇧05 Configuri

What	*nameofinstance*.xml
Description	Configuration for the new version of the transformation modul
schema location	http://fews.wldelft.nl/schemas/version1.0/transformationModul
Entry in ModuleDescriptors	<moduleDescriptor id="TransformationModule"> <description>Transformation Module</description> <className>nl.wldelft.fews.system.plugin.transformationmodu </moduleDescriptor>

Contents

Learning Objectives

By the end of this module, you will have met the following learning objectives:

1) Understand the roles of transformations in Delft-FEWS
2) Be able to configure a basic transformation, run it in a module and view the results
3) Know where to find further information for transformations of the WIKI.

FEWS Concept Review - Modules

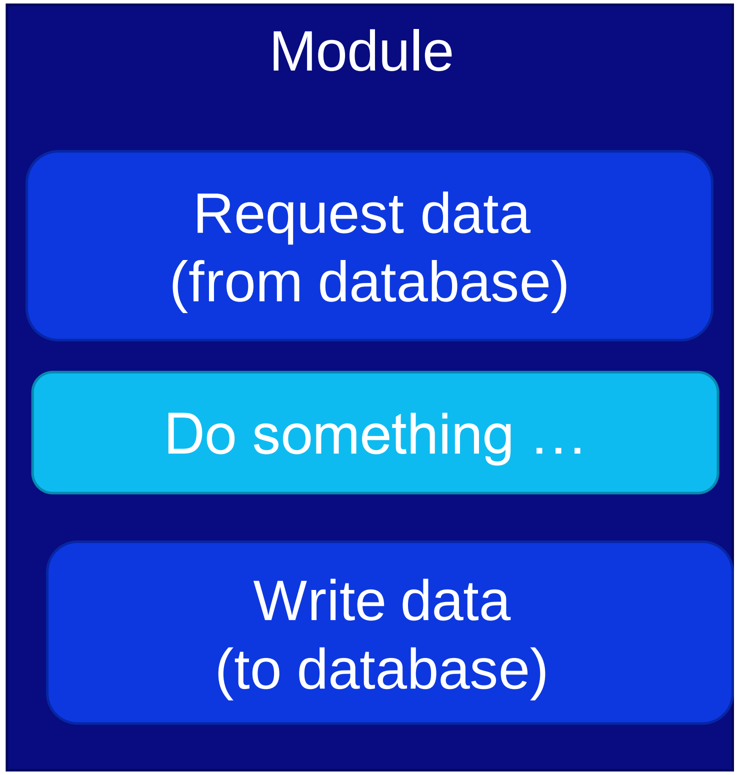
- A Delft-FEWS module is an internal/external module that does ‘something’ with time series**:**
- Requests time series from the database
- Processes the data
- Writes processed time series to database

•Example Import module

- Module imports time series from files, database or server
- Convert the series to internal FEWS ID’s (locations, parameters)
- Write time series to FEWS database

•Example Aggregation module (Transformation)

- Module retrieves time series from FEWS database
- Aggregates the time series to different time step following configured rules
- Write aggregated time series to FEWS database



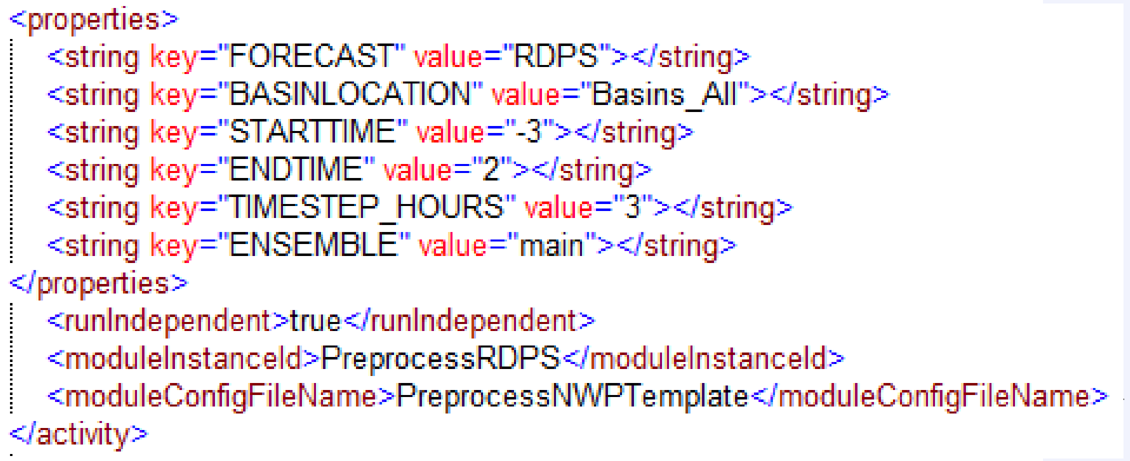
Deltares

ModuleInstances & ModuleInstanceDescriptors

- (Data processing) modules are configured in a ModuleConfigFile
  - holds the instructions to retrieve data, do something and store data

- ModuleInstances are instances of a module as it is called in the workflow
  - Identified by ModuleInstanceId
  - registered in \RegionConfigFiles\ModuleInstanceDescriptors.xml

- One ModuleConfigFile can act as a template for multiple moduleInstances
  - $PROPERTIES$ can be used to make time series explicit
  - $PROPERTIES$ can be provided by the workflow
  - $PROPERTIES$ are resolved at run-time



```
<properties>
   <string key="FORECAST" value="RDPS"></string>
   <string key="BASINLOCATION" value="Basins_All"></string>
   <string key="STARTTIME" value="-3"></string>
   <string key="ENDTIME" value="2"></string>
   <string key="TIMESTEP_HOURS" value="3"></string>
   <string key="ENSEMBLE" value="main"></string>
</properties>
   <runIndependent>true</runIndependent>
   <moduleInstanceId>PreprocessRDPS</moduleInstanceId>
   <moduleConfigFileName>PreprocessNWPTemplate</moduleConfigFileName>
</activity>
```

Deltares

Data manipulation: Transformation Module

- Transformation modules: <u>Workhorse</u> of DELFT-FEWS

- Configured in Config/ModuleConfigFiles
- One configuration file can define multiple transformations
- Each configuration file must be registered in ModuleInstanceDescriptors.xml

- Input and outputs are always time series or coefficients

- Input/output location sets may vary, but output locations always need to be correctly referenced to input locations, like grids to scalar.

- Processing can be made conditional based on:
  - value range
  - date (before, in-between, after)

Data manipulation: Transformation Module

•Long list of transformation functions can be selected

- Accumulation: sum,...
- Aggregation: accumulative, ...
- Disaggregation : accumulative, ...
- DischargeStage/StageDischarge: table, ...
- InterpolationSerial: block, default, extrapolate, linear
- InterpolationSpatial: avg, closestDistance, inverseDistance, thiessen,....
- Lookup: 2D, ...
- Merge: selectLocation, selectDataSource, simple, ...
- Profile: timeseries,...
- Sample: equidistant, nonequidistant, ...
- Statistics(RelatedLoc.,Ensembles, ...): max, mean, min, percExceedence
- UserDefined: simple expression, ...

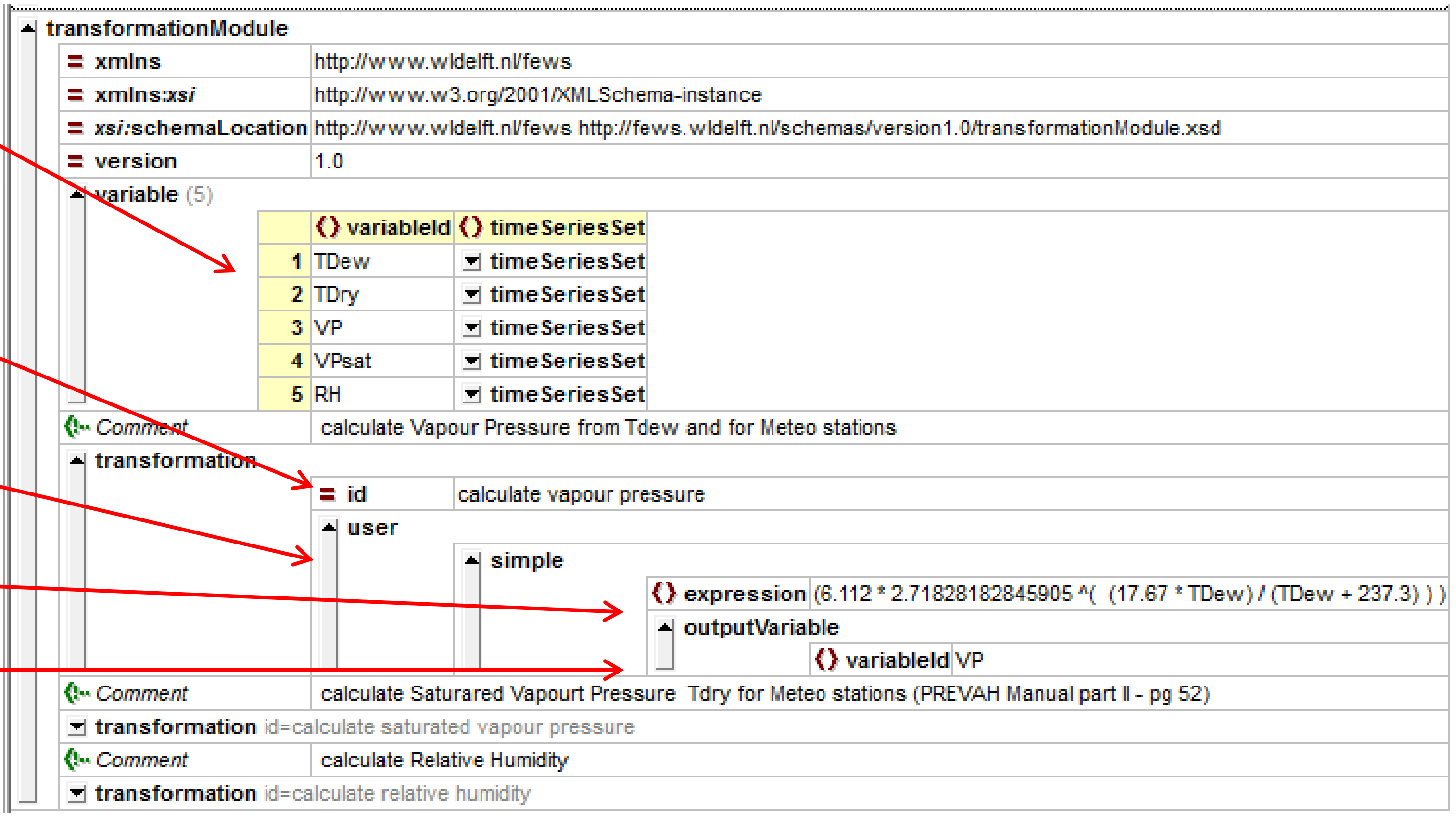
- Google “Delft-FEWS Transformations” for complete documentation

Deltares

Typical Transformation Module Instance

•User function with a user defined function

- Variables (timeserieSets)
- Transformation Id
- Function Type
- Expression
- Output variable



```
transformationModule
  = 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/transformationModule.xsd
  = version               1.0
  variable (5)
         () variableId   () timeSeriesSet
      1  TDew               timeSeriesSet
      2  TDry               timeSeriesSet
      3  VP                 timeSeriesSet
      4  VPsat              timeSeriesSet
      5  RH                 timeSeriesSet
  Comment                 calculate Vapour Pressure from Tdew and for Meteo stations
  transformation
      = id                calculate vapour pressure
      user
          simple
              () expression  (6.112 * 2.71828182845905 ^(  (17.67 * TDew) / (TDew + 237.3) ) )
              outputVariable
                  () variableId  VP
  Comment                 calculate Sarurared Vapourt Pressure  Tdry for Meteo stations (PREVAH Manual part II - pg 52)
  transformation id=calculate saturated vapour pressure
  Comment                 calculate Relative Humidity
  transformation id=calculate relative humidity
```

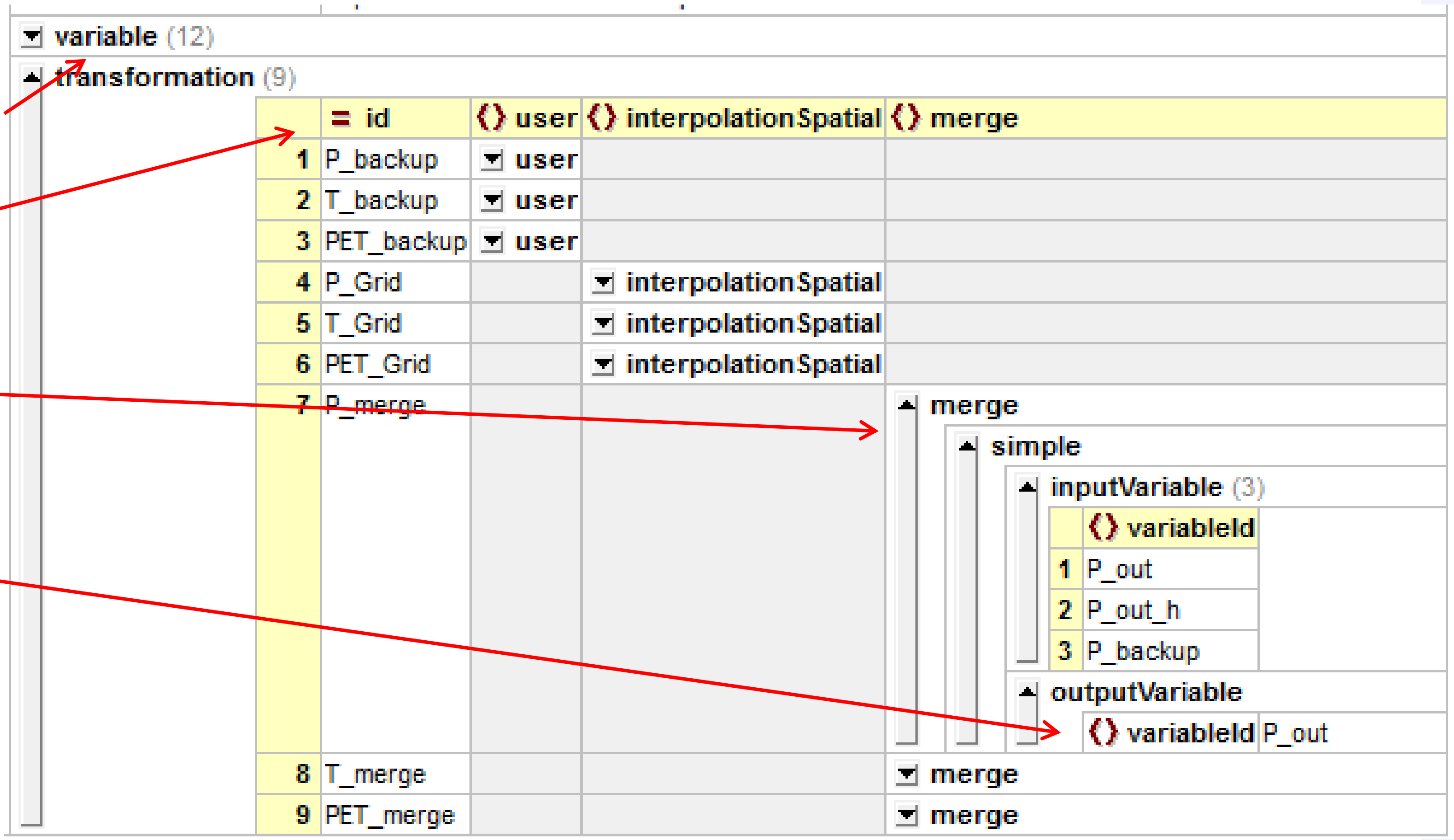
- One transformation module instance can contain multiple functions

Deltares

Typical Transformation Module Instance

•Pre-defined functions (i.e. Merge or Data Hierarchy)

- Variables (timeserieSets)
- Transformation Id
- Function Type
- Output variable



variable (12)

transformation (9)

	id	user	interpolationSpatial	merge
1	P_backup	user		
2	T_backup	user		
3	PET_backup	user		
4	P_Grid		interpolationSpatial	
5	T_Grid		interpolationSpatial	
6	PET_Grid		interpolationSpatial	
7	P_merge			merge
8	T_merge			merge
9	PET_merge			merge

7 P_merge – merge:

merge
  simple
    inputVariable (3)

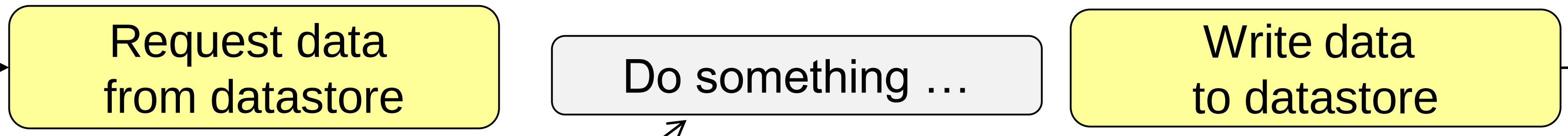
	variableId
1	P_out
2	P_out_h
3	P_backup

    outputVariable
      variableId P_out

Deltares

Transformation – Input / Output

```
<!--Input-->
<variable>
        <variableId>Gridded_parameter</variableId>
                   <timeSeriesSet>
                   <moduleInstanceId>$INPUTMODULEINSTANCE$</moduleInstanceId>
                   <valueType>grid</valueType>
                   <parameterId>$PARAMETER$</parameterId>
                   <locationId>$INPUTGRID$</locationId>
                   <timeSeriesType>$TIMESERIESTYPE$</timeSeriesType>
                   <timeStep unit="hour" multiplier="$TIMESTEP$"/>
                   <readWriteMode>read complete forecast</readWriteMode>
                   <ensembleId>$ENSEMBLE$</ensembleId>
          </timeSeriesSet>
</variable>
```



```
<transformation id="station_parameter">
    <interpolationSpatial>
        <closestDistance>
                    <inputVariable>
                       <variableId>Gridded_parameter</variableId>
                    </inputVariable>
                    <distanceGeoDatum>$DISTANCEGEODATUM$</distanceGeoDatum>
                    <outputVariable>
                       <variableId>Station_parameter</variableId>
                    </outputVariable>
       </closestDistance>
    </interpolationSpatial>
</transformation>
```

```
<variable>
        <variableId>Station_parameter</variableId>
        <timeSeriesSet>
                    <moduleInstanceId>$MODULE_INSTANCE_ID$</moduleInstanceId>
                    <valueType>scalar</valueType>
                    <parameterId>$PARAMETER$</parameterId>
                    <locationSetId>$STATIONLOCATIONS$</locationSetId>
                    <timeSeriesType>$TIMESERIESTYPE$</timeSeriesType>
                    <timeStep unit="hour" multiplier="$TIMESTEP$"/>
                    <readWriteMode>read complete forecast</readWriteMode>
                    <ensembleId>$ENSEMBLE$</ensembleId>
         </timeSeriesSet>
</variable>
```

Deltares

Module Summary

- Processing of data in Delft-FEWS is done by modules, and more specifically transformations
- Modules and Transformations can be given any name, so be descriptive.
- Reading and writing of data from the database is done with TimeSeriesSets
- Transformations can be quite simple, and quite powerful.
- Multiple transformations can be in one module, and multiple modules can be in one workflow

Next Steps

- Now that we know how data is imported and processed in Delft-FEWS, we'll now look at how it is displayed.
- Data can be displayed in Plots and in the Spatial Display
- We'll first look at Plot Displays, that have a rich functionality both to view data on the Fly, and for pre-defined plots.
- Our configured, this provides the simplest way to view data.

Additional Resources

- Google “Delft-FEWS WIKI”
- Google “Delft-FEWS Configuration Guide”
- Google “Delft-FEWS Forum”
- Email fews-pm@Deltares.nl