01 Structure of a Delft-FEWS Configuration

Introduction

The main part of a Delft-FEWS configuration is defined in sets of XML files. In this section the different parts of the configuration are introduced. An understanding of these different parts of the configuration is required before attempting configuration of a Delft-FEWS system.

Stand alone client with configuration on local filesystem

Typically, a new Delft-FEWS configuration is first setup in a Stand alone application, using a configuration setup on the local file system. In this case, the configuration is defined by a fixed set of directories, each containing different parts of the configuration. These directories are all contained under the *Config* directory.

Operator client configuration in local datastore and central database

When distributing the application, the configuration files are instead shared via a local datastore or the central database. The local datastore or the central database, stores the same local filesystem configuration directory structure in dedicated SQL tables, each table containing different configuration directory. The configuration in the central database belongs to a dedicated Delft-FEWS implementation of a particular forecasting system. In the live system situation the contents of the database will be shared between all operator clients and forecasting shell servers in the system, and is therefore expected to be identical in all parts of the system. In the central database active versions of configuration items have *synchLevel 11* whereas inactive configuration files have *synchLevel 0*. The *ConfigRevisionSets* table defines the active configuration.

When initiating the Delft-FEWS application, it will look for configuration stored in the local datastore or in the filesystem. If both are found, then the system might ask the user to choose which to use. If neither is found then an appropriate error message is issued and the system will stop.

RootConfig files

A small set of XML files referred to as the root configuration files. This root configuration is required to identify for example if the particular instance of Delft-FEWS is operating in stand-alone mode or as an operator client, or on windows or linux. Some configuration files may differ between operator client and/or forecasting shell server. These files have a filename with a meaningful prefix such as 'oc_windows_' or 'fss_linux_'.

Elements of the configuration

The two tables below provide an overview of the configuration elements of Delft-FEWS. Table 1 Overview of different configuration items contained either in the config directory or in the database:

Configuration Item	Directory on filesystem	Table name in database	Single/ Multiple	Documentation
Transformation coefficient sets for specifying ranges or conditions in locations in specific time periods.	CoefficientSetsFiles	CoefficientSets		
Cold states for modules. Zip file containing model specific data exported by GA usually before running a model.	ColdStateFiles	ColdStates	Multiple	
CorrelationEvents describing discharges, levels or other obtained observations for historical floods.	CorrelationEventSetsFiles	CorrelationEventSets		
Definition of layout of user displays, including What-if scenarios, Grid Display, etc.	DisplayConfigFiles	DisplayConfigurations	Multiple	07 Display Configuration
Flag conversions between external sources (e.g. telemetry, modules) and flags used in Delft-FEWS internally.	FlagConversionsFiles	FlagConversions	Multiple	08 Mapping Id's flags and units
Icons used in main map display and button bar	IconFiles	Icons	Single	
Mapping definition of ID's between external sources (e.g. telemetry, modules) and ID's used in Delft-FEWS internally.	IdMapFiles	IdMaps	Multiple	
Map layers (shape files) used in main map display and spatial interpolation	MapLayerFiles	MapLayers	Single	

Module configuration for handling data and running forecasting models	ModuleConfigFiles	ModuleInstanceConfigs	Multiple	
Zipped files containing datasets for modules used by the forecasting system.	ModuleDataSetFiles	ModuleInstanceDatasets	Multiple	09 Module datasets and Module Parameters
Definition of module parameters stored in Delft-FEWS	ModuleParFiles	ModuleParameters	Multiple	09 Module datasets and Module Parameters
Files for the PiClient.	PiClientConfigFiles	PiClientConfigurations	Multiple	
Triggers export/import for modules.	PiServiceConfigFiles	PiServiceConfigurations	Multiple	
Regional configuration, including all locations, location sets, parameters, several types of descriptors, etc..	RegionConfigFiles	RegionConfigurations	Single	04 Regional Configuration
Images used in reports.	ReportImageFiles	ReportImageFiles	Single	09 Report Module
HTML, css, js template files used in creating HTML reports for use on the web server.	ReportTemplateFiles	ReportTemplates	Multiple	09 Report Module
Operating system configuration, clientConfiguration, global properties and patch.jar for supporting automated updates.	RootConfigFiles	RootConfigFiles	Multiple	01 Root Configuration Files
System configuration items, including the plug-ins available to the system, definition, icons etc.	SystemConfigFiles	SystemConfigurations	Single	03 System Configuration
CorrelationEvent travel times describing travel time for events from location to location.	TravelTimesFiles	CorrelationTravel Times		
Unit conversions between external sources (e.g. telemetry, modules) and units used in Delft-FEWS internally.	UnitConversionsFiles	UnitConversions	Multiple	08 Mapping Id's flags and units
Workflow configuration for running sequences of modules.	WorkflowFiles	WorkflowFiles	Multiple	06 Workflow Configuration

Naming conventions for defining module config files

For complex forecasting systems the number of configuration files can be very large. This is particularly the case for the module config files, and because the names of these are used in TimeSeriesSets for storing and retrieving data, the names given should be chosen logically. Before configuring large numbers of XML files it is wise to:

1. define a naming convention and use this throughout. An example of such a convention is where a number of steps are used to process data prior to running a model. For instance, a forecast model run for the HBV model in the Rhine may be defined in a module called *HBV_Rhine_Forecast.xml.* Data processing steps such as an interpolation module may then be called *HBV_Rhine_ForecastInterpolate.xml* and a data merge module *HBV_Rhine_ForecastMergeInputs.xml*. This clearly indicates the association between modules and brings structure to the configuration.
2. when multiple almost identifcal configuration files would be required, it is recommended wherever possible to use template configuration files. These template configuration files contain $ tagged variables that can be filled in as as properties when referred to by other XML files.

XML Schemas and schema validation

Each configuration item contained in an XML file must be formatted as specified in an appropriate XML schema (XSD file). Validating against the schemas is an important step in configuring Delft-FEWS, as the XML validation makes sure the syntax of the configuration made is correct. There are two types of configuration in Delft-FEWS. In the first set, for each different schema type, only one default configuration file may be used and the name of the configuration file is unique. For the second set of configuration, multiple configuration types may be available for a specific schema. The names of these may be defined by the user. An XML file contained in the regional configuration element is then used to register these XML files with a user specified name to the system, and identify the type of configuration. This file is referred to as a *descriptor* file. Table1 identifies for which type of configuration a single files per type is allowed and for which multiple instances for each type of configuration may exist.

Configuration versioning file naming conventions (now obsolete)

NB. The "default" versioning name convention described here is not required / recommend anymore for new configurations. In general it is good practice to manage config versioning using SVN tooling, see https://en.wikipedia.org/wiki/Apache_Subversion.

For each of the configurations managed by Delft-FEWS in either the database or on the file system as described above, various versions of configuration may exist. Configurations that are active and used as a default can be identified both in the file system and in the database. On the file system an optional naming convention is introduced to identify which of the possible multiple versions are used as a default. The naming convention for the default version:

<Name of XML configuration file>SPACE<Version number>SPACE<default>.xml

Another version of configuration will have a different version number. The <default> item is omitted. Examples:

example	description
Explorer 1.00 default.xml	A "default" version of the configuration settings for the FEWS Explorer
Explorer 2.00.xml	A second version that is not made active, since it has not the "default" tag.