ClientConfig XML File for Operator Client and Forecasting Shell Servers - 2018.02 and later

clientConfig.xml file(s)

From Delft-FEWS 2018.02 onwards, the different *clientConfig.xml* files have an important place in the Delft-FEWS root configuration. Different variants (for different components) of clientConfig.xml files exist and will be explained below.

The clientConfig.xml consist of the following elements:

title

Optional element for setting the title of your application/component

clientType

Required element for defining the type. Enumeration of options includes;

- Forecasting Shell
- Operator Client
- Stand Alone

otherRootConfigFiles

Required element. One or more references to the required root configuration files which should always be downloaded when an OC or FSS starts. For an OC the following root configuration files are important to mention here: patch.jar, oc_global.properties, splash-screen reference and oc_clientConfig.xml. See below mentioned image. Important remark is that when the OC starts, it will be checked if these files are present. If not, they will be downloaded. If they are already there, it will be checked it is the correct active version as in the central database. If the files on the file systems are not the active version, they will be downloaded and overwritten.

connection

Optional element to define the connections to the Master Controller. Only required if you have more than 1 Master Controller connections (multi MC environments).

If there is only 1 Master Controller, the connection details are already within the *shortcuts* (TO DO: INSERT LINK TO EXPLANATION) you have created when deploying/installing the system.

databaseServer (+ url / user / password)

Required elements for setting the details of the database server if you want to specify those in the **<connection>**. See example below

Example of a clientConfig.xml with more than 1 connection (multi MC environment)

```
<clientConfiguration 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/clientConfig.xsd">
        <title>HyFS-RM2018-OC</title>
        <clientType>Operator Client</clientType>
        <otherRootConfigFiles>
                <name>patch.jar</name>
                <name>fews-splash.jpg</name>
                <name>oc_delft_global.properties</name>
        </otherRootConfigFiles>
        <connection id="mc00_postgresql">
                <databaseServer>
                        <url>jdbc:postgresql://myhost:5432/mc00</url>
                        <user>username</user>
                        <password>######</password>
                </databaseServer>
        </connection>
        <connection id=" mc01_oracle">
                <databaseServer>
                        <url>jdbc:oracle:thin:@myhost:1521:mc01</url>
                        <user>rm201802</user>
                        <password>######</password>
                </databaseServer>
        </connection>
        <connection id="mc02_sqlserver">
                <databaseServer>
                        <url>jdbc:sqlserver://myhost:1433;DatabaseName=mc02</url>
                        <user>fewsuser</user>
                        <password>######</password>
                </databaseServer>
        </connection>
        <jvmOption>-Xmn1000m</jvmOption>
        <jvmOption>-Xmx1500m</jvmOption>
        <autoExportModuleDataSet name="HyFS_Help" exportDir="products"/>
        <autoExportModuleDataSet name="AUS_Products" exportDir="products"/>
        <autoExportModuleDataSet name="QLD_Products" exportDir="products"/>
        <autoExportModuleDataSet name="NSW_Products" exportDir="products"/>
        <autoExportModuleDataSet name="SA_Products" exportDir="products"/>
        <autoExportModuleDataSet name="WA_Products" exportDir="products"/>
        <autoExportModuleDataSet name="NT_Products" exportDir="products"/>
        <autoExportModuleDataSet name="TAS_Products" exportDir="products"/>
        <autoExportModuleDataSet name="VIC_Products" exportDir="products"/>
 </clientConfiguration>
```

jvmOption

Optional element to specify additional java VM options (which were present in the launcher *ini file in earlier versions of Delft-FEWS). In short: all -D and -Xmx options can be configured in individual jvmOptions, example see below.

- -Xms
- -Xmx
- -DautoRollingBarrel=false

localCacheSizeMB

Local cache size (in MB, default is 500) for downloaded timeseries from central database when using Direct Database Access (or DDA) or from the Open Archive Server. This is NOT a feature when you are using synchronization to download timeseries locally to create a LocalDataStore (or LDS). If omitted, the default value of 500MB will be used.

localDataStoreFormat

Use this option to generate a local datastore of a specific format. Will (only) be created when no local datastore is present yet and only in case when a synchProfile is present.

Options are: Derby (default), Firebird or HyperSQL.

logging

Option to generate some extra system logging.

The following example demonstrates how event logging is enabled for an OC or FSS by addition of a *logging* section to the clientConfig.xml.

```
<clientConfiguration 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/clientConfig.xsd">
<clientType>Operator Client</clientType>
  ...
 <logging>
        <debugEnabled>false</debugEnabled> <!--since 2018.02-->
        <rollingTotalSizeMB>5</rollingTotalSizeMB> <!--since 2018.02, when a log file is reaching 1 MB a new
log file is created with the next sequence number. Only the last 5 log files are kept-->
        <windowsEventLogEnabled>true</windowsEventLogEnabled>
        <linuxSyslogFacility>local6</linuxSyslogFacility>
    </logging>
</clientConfiguration>
```

If enabled, the event logging sents a subselection of all log messages that were logged via log4j to the operating system event log.
The selection includes

- all errors with an event code (an event-code is the first substring in the log message plus a semi-colon).
- all messages with at least level INFO and the prefix or postfix of the event code contains "Syslog:" or "Syslog." (case insensitive)

If for instance a custom adapter is required to log to this facility, the adapter should log using log4j with a Syslog prefix in the message, e.g.

```
log.info("Syslog.Adapter: imported file" + filename);
```

coldStatesDirectory

Since 2020.02. A string path to a accessible directory which holds the cold states. When specified all the cold states are stored in this directory instead of in the database. Only use this option when your cold states are very large (500MB+). First remove all the existing cold states with the config manager from the database.

warmStatesDirectory

A string path to a accessible directory which holds the warmstates of certain models. In their corresponding general adapter run (model run), a forecasting model can write warmstates to a location on the server (instead of storing it in the database).

proxyAutoConfigScriptContent

Use this option when you want to use your own WPAD script. Use a CDATA block around your script.

proxyAutoConfigScriptUrl

Use this option if a ProxyAutoConfig (PAC) file is located on e.g. your intranet page.

proxyAutoConfig configuration examples

```
<clientConfiguration 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/clientConfig.xsd">
        <localDataStoreFormat>Firebird</localDataStoreFormat>
        <proxyAutoConfigScriptContent>
<![CDATA[*/
                function FindProxyForURL(url, host)
                {
                return "DIRECT";
                }
/*]]>
        </proxyAutoConfigScriptContent>

        <! -- OR USE -->
        <proxyAutoConfigScriptUrl>http://xxxxxxxxxx/wpad.pac</proxyAutoConfigScriptUrl>

</clientConfiguration>
```

autoExportModuleDataSet

Optional element with references to the ModuleDataSetFiles that should be extracted when an OC or FSS starts. Whenever an FSS initiates a workflow, it verifies if it should (re-)extract these ModuleDataSetFiles. This verification is done based on comparing checksums (database contents versus local checksum which is logged in /temp/moduleDataSetsCheckSums). If a change is identified, files will be deleted and then re-extracted. Likewise, if no extracted files are found, a fresh extract will be performed.

This option is mostly valid for FSSs since they are running the models. In fact: the complete structure on an FSS underneath the "/Modules/" folder should (ideally) be stored in one or more ModuleDataSet files. Only in this way, FSSs can be used for scaling up/down since its content is generated automatically. See below mentioned image with a number of references to ModuleDataSet files.

In case of using the new Web Browser Display this option should also be used to distribute the required libraries for an OC.

The exportDir is relative to %REGION_HOME% unless otherwise specific. The name references the moduleDataSet file-name without the zip-extension. It is not required register to this file-name as a moduleInstanceId.

```
<autoExportModuleDataSet name="VIC_Products" exportDir="products"/>
```

Unzips the VIC_Products.zip to %REGION_HOME%/products (d:\fews\fss\1\products) (/opt/fews/fss/1/products)

truststore

When the ssl certificate issuer used by a https web host mentioned in the FEWS configuration isn't bundled with the (Amazon) java run time you need to create a truststore file. Unfortunately this will not work when the certificate is used by the database connection itself.

Create this truststore file with

- (2018.02 build 87948 and later) a FEWS **standalone** (so not at an OC) - F12 convert convert certificate file to truststore file.
- (older builds) the keytool.exe located in the jre/bin dir. Use a cmd prompt with something like: <location of fews bin>\jre\bin\keytool.exe -import -v -alias root -file <certificate file> -keystore <name>.truststore -storepass <pass>

Check the contents of a truststore with the keytool (cmd: keytool -list -v -keystore <name>.truststore). Look for

- "Owner: CN=" with the hostname, and
- SubjectAlternativeName, with alternative DNSNames

Add the truststore file to the otherRootConfigFiles so it is automatically downloaded on a OC/FSS/PI.

synchProfile

A synchProfile can be defined if you would like to create an **OC with LocalDataStore**. By default the connection will be Direct Data Access (DDA).

Per synchProfile you can define what should be synched. Most elements are compulsory and can be set to enabled (true/false). Within the timeseries, warmstates, logentries and thresholdevents you can configure a certain maximum age. See config example below. Ensure that you include all relevant synchLevels in your application. See also the definition of default synchLevel here.

Configuration Example for a synchProfile section

```
<clientConfiguration 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/clientConfig.xsd">
    <title>HyFS-RM2018-OC</title>
    <clientType>Operator Client</clientType>
    <otherRootConfigFiles>
        <name>patch.jar</name>
        <name>fews-splash.jpg</name>
        <name>oc_delft_global.properties</name>
    </otherRootConfigFiles>
    <connection id="mc00_postgresql">
        <databaseServer>
            <url>jdbc:postgresql://dl-081.xtr.deltares.nl:5432/core00</url>
            <user>username</user>
            <password>######</password>
        </databaseServer>
        <synchProfileId>full</synchProfileId>
        <synchProfileId>minimal</synchProfileId>
    </connection>
    <connection id=" mc01_oracle">
        <databaseServer>
            <url>jdbc:oracle:thin:@pl-or001.xtr.deltares.nl:1521:rm201802</url>
            <user>rm201802</user>
            <password>######</password>
        </databaseServer>
        <synchProfileId>full</synchProfileId>
        <synchProfileId>minimal</synchProfileId>
    </connection>
    <connection id="mc02_sqlserver">
        <databaseServer>
            <url>jdbc:jtds:sqlserver://dw-ms002.xtr.deltares.nl:1433;DatabaseName=roadmapmc01;useCursors=false;
sendStringParametersAsUnicode=false</url>
            <user>fewsuser</user>
            <password>######</password>
        </databaseServer>
        <synchProfileId>full</synchProfileId>
        <synchProfileId>minimal</synchProfileId>
    </connection>
    <synchProfile id="full">
        <xmlConfig enabled="true" name="Default xml config" synchLevel="11"/>
        <coldStates enabled="true" name="Default cold states" synchLevel="11"/>
        <moduleDataSets enabled="true" name="Default module data sets" synchLevel="11"/>
        <mapLayers enabled="true" name="Default module data sets" synchLevel="11"/>
        <icons enabled="true" name="Default icons" synchLevel="11"/>
        <reportTemplates enabled="true" name="Default report templates" synchLevel="11"/>
        <reportImages enabled="true" name="Default report images" synchLevel="11"/>
        <!-- this is just an example and does not include the full list of possible synchLevels!! -->
        <continuousTimeSeries enabled="true" name="Telemetry" synchLevel="1" maxAge="10" unit="day"/>
        <continuousTimeSeries enabled="true" name="Manual" synchLevel="5" maxAge="10" unit="day"/>
        <continuousTimeSeries enabled="true" name="Astronomical and climatological" synchLevel="4" maxAge="
1000" unit="day"/>
        <continuousTimeSeries enabled="true" name="Small external forecast grids" synchLevel="6" maxAge="10"
unit="day"/>
        <continuousTimeSeries enabled="true" name="Large external forecast grids" synchLevel="16" maxAge="10"
unit="day"/>
        <forecastTriggeredTimeSeries name="Simulated forecast time series" enabled="true" synchLevel="0"/>
        <warmStates enabled="true" name="Warm states" maxAge="10" unit="day"/>
        <logEntries enabled="true" name="Log Entries" maxAge="10" unit="day"/>
        <thresholdEvents enabled="true" name="Threshold Events" maxAge="10" unit="day"/>
    </synchProfile>
    <synchProfile id="minimal">
        <xmlConfig enabled="true" name="Default xml config" synchLevel="11"/>
        <coldStates enabled="true" name="Default cold states" synchLevel="11"/>
        <moduleDataSets enabled="false" name="Default module data sets" synchLevel="11"/>
        <mapLayers enabled="true" name="Default module data sets" synchLevel="11"/>
        <icons enabled="true" name="Default icons" synchLevel="11"/>
        <reportTemplates enabled="false" name="Default report templates" synchLevel="11"/>
        <reportImages enabled="false" name="Default report images" synchLevel="11"/>
        <continuousTimeSeries enabled="true" name="Telemetry" synchLevel="1" maxAge="1" unit="day"/>
        <continuousTimeSeries enabled="true" name="Manual" synchLevel="5" maxAge="1" unit="day"/>
        <continuousTimeSeries enabled="true" name="Astronomical and climatological" synchLevel="4" maxAge="
```

```
1000" unit="day"/>
        <continuousTimeSeries enabled="false" name="Small external forecast grids" synchLevel="6" maxAge="10"
unit="day"/>
        <continuousTimeSeries enabled="false" name="Large external forecast grids" synchLevel="16" maxAge="10"
unit="day"/>
        <forecastTriggeredTimeSeries name="Simulated forecast time series" enabled="true" synchLevel="0"/>
        <warmStates enabled="false" name="Warm states" maxAge="10" unit="day"/>
        <logEntries enabled="true" name="Log Entries" maxAge="1" unit="day"/>
        <thresholdEvents enabled="true" name="Threshold Events" maxAge="1" unit="day"/>
    </synchProfile>
</clientConfiguration>
```

externalTables

Here you can indicate whether you are using an 'external' (= non-blobbed) database for: Parameters, Locations, Qualifiers and External Historical Timeseries. You have to specify the connection details to this (database) server here.

Example ClientConfiguration

Example Configuration: (minimal) Stand Alone sa_clientConfig.xml

```
<clientConfiguration 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/clientConfig.xsd">
        <title>my Delft-FEWS SA</title>
        <clientType>Stand alone</clientType>
    <jvmOption>-Xmx1500m</jvmOption>
</clientConfiguration>
```

Example Configuration: (minimal) Operator Client oc_clientConfig.xml

```
<clientConfiguration 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/clientConfig.xsd">
        <title>my Delft-FEWS OC</title>
        <clientType>Operator Client</clientType>
        <otherRootConfigFiles>
                <name>oc_global.properties</name>
                <name>patch.jar</name>
                <name>help.pdf</name>
                <name>fews-splash.jpg</name>
        </otherRootConfigFiles>
    <jvmOption>-Xmx1500m</jvmOption>
        <!-- ONLY IF MODULEDATASETS NEED TO BE EXTRACTED ON STARTUP -->
        <autoExportModuleDataSet name="X" exportDir="X"/>
</clientConfiguration>
```

Example Configuration: (minimal) Forecasting Shell fss_clientConfig.xml

```
<clientConfiguration 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/clientConfig.xsd">
        <title>my Delft-FEWS FSS</title>
        <clientType>Forecasting Shell</clientType>
        <otherRootConfigFiles>
                <name>fss_global.properties</name>
                <name>patch.jar</name>
        </otherRootConfigFiles>
    <jvmOption>-Xmx1500m</jvmOption>
 <!-- ONLY IF MODULEDATASETS NEED TO BE EXTRACTED ON STARTUP -->
        <autoExportModuleDataSet name="X" exportDir="X"/>
</clientConfiguration>
```