

# 01 Root Configuration Files for Operator Client

## Root configuration files

Root configuration files define the behaviour of DELFT-FEWS on the local machine. These files are synchronised in the live system environment, nor are they available in the database. The files must be installed locally with the DELFT-FEWS system.

The root configuration items include;

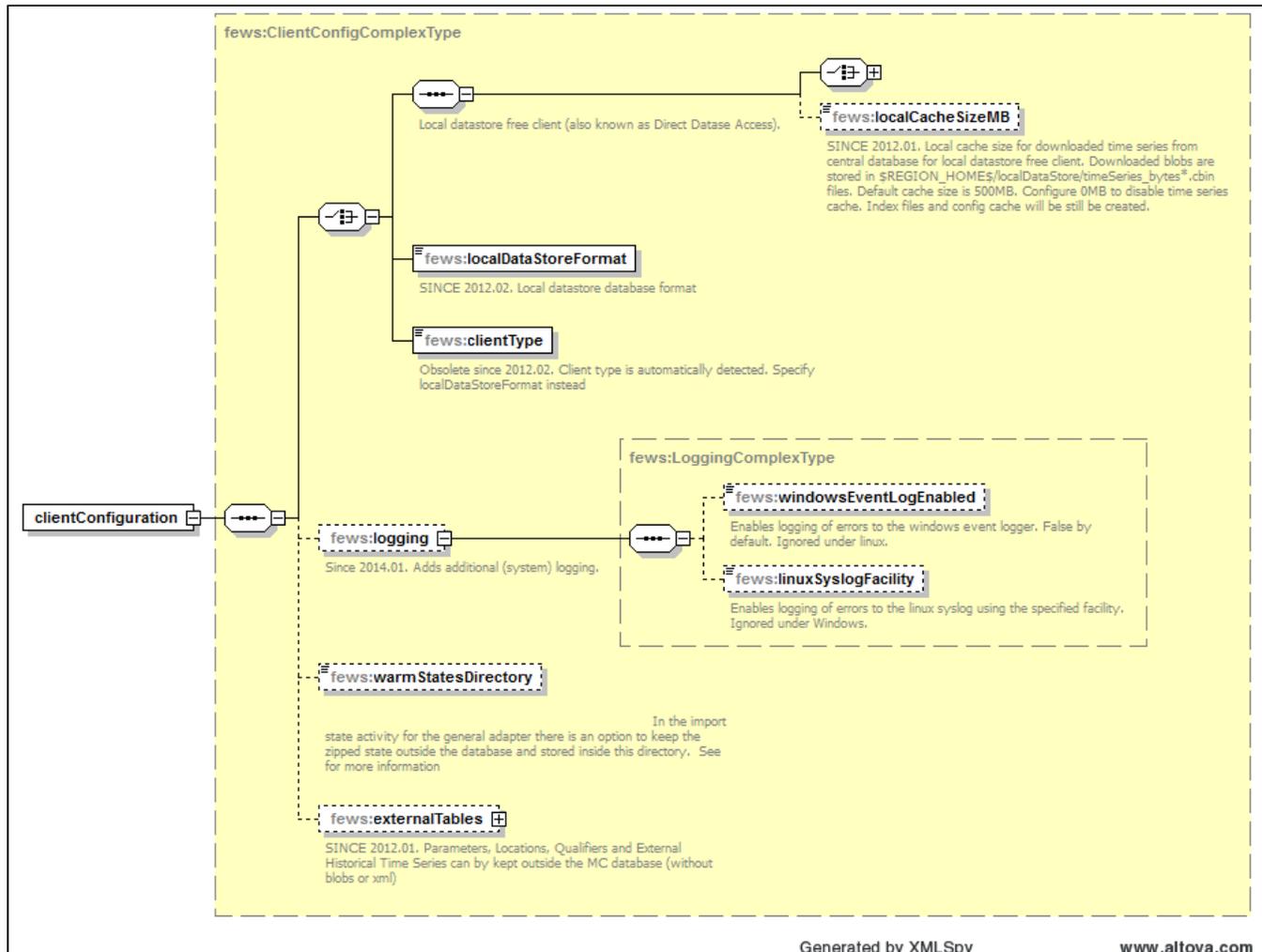
- clientConfig
- logConfig
- RollingBarrel\_Database (this file should not be changed and is not described)
- synchConfig
- synchChannels
- synchProfiles
- [global.properties](#)

**!** The clientType of a system may be changed from *Operator Client* to *Stand alone*. Under **no** conditions should a system that was once StandAlone be changed to an Operator Client system. This may only be done if the local data store is first deleted.

## clientConfig

The clientConfig File determines if the instance of DELFT-FEWS is to run as a stand alone system, or if it is to connect to the master controllers defined below.

Since 2011.01 the clientConfig file is no longer required for stand alone. Stand alone is the default when the client config is missing



Elements of the clientConfig configuration

## clientType

Definition of the client type. Enumeration of options includes;

- Operator Client
- Stand Alone
- Computational Framework (Since 2020.01)  
Stand Alone with additional features dedicated to local computations with make it impossible to change the system to an OC later.

## logging

Since 2014.01 it is possible to send errors from the OC/FSS/standalone client to the windows or linux event log. This option is not enabled by default.

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

```
<?xml version="1.0" encoding="UTF-8"?>
<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 sends 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);
```

Per default the Operator Client logs key events for logins to this facility.

```
Delft-FEWS Operator Client: userName=User mcId=MC INFO - RemoteSession.login - Login.Started: Login to Master
Controller
Delft-FEWS Operator Client: userName=User mcId=MC INFO - RemoteSession.login - Login.Finished: Successfully
logged in to Master Controller: USBPMC00. Session ID USBPMC00:000017435
Delft-FEWS Operator Client: userName=User mcId=MC INFO - RemoteSession.logout - Logout.Finished: Successfully
logged out from Master Controller: USBPMC00
Delft-FEWS Operator Client: userName=User mcId=MC INFO - RemoteSession.logout - Login.Failed: Cannot Log to
Master Controller: USBPMC00
```

Sample linux configuration (Redhat based system) (nippet of modifications to /etc/rsyslog.conf) that sends errors and logins to /var/logs/fews.log.

```
# Provides UDP syslog reception
$ModLoad imudp
$UDPServerRun 514

local6.* /var/log/fews.log
```

## LogConfig

*To be completed*

## synchConfig

Looks something like this for an OC:

```

<?xml version="1.0" encoding="UTF-8"?>
<fews-master-config 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
/synchronisationConfiguration.xsd">
  <queueconnection>
    <factory jndi="ConnectionFactory"/>
  </queueconnection>
  <defaultMcId>MC00</defaultMcId>
  <mc id="MC00">
    <jndicontext factory="org.jnp.interfaces.NamingContextFactory" provider="jnp://localhost:1099"
prefixes="org.jboss.naming:org.jboss.interfaces"/>
    <queue>
      <root jndi="TEST/MC00"/>
      <synch jndi="External/JMSQueue/OCIncoming" timeout="10"/>
    </queue>
  </mc>
  <synchronisation>
    <messaging maxrecords="1000" maxlobdata="30000000"/>
    <processor maxlistsize="500000"/>
    <schema location="nl/wldelft/fews/master/data/synchdata/synchronisation_schema.xsd"/>
  </synchronisation>
  <login timeout="10" />
</fews-master-config>

```

The synchConfig does not normally require editing.

The only setting that may demand editing is the <login timeout="10" />, which controls the timeout for a login attempt of the OC on the MC (in seconds). (This element may be absent, in which case the timeout is 10secs)

It may be needed to extend this timeout if the JMS server is very busy (very many clients starting up and synchronising at the same time, e.g. when all the PCs for a workshop are all starting up at the same time).

Note: the xml config can only extend the time out from the default 10 secs. Settings less than 10 secs are ignored)

## synchProfiles

The file synchProfiles.xml contains several different profiles for fine-grained control over the synchronisation with the database.

The following profiles

Full	Profile for synchronising fully between the Operator Client and the Master Controller
Minimal	Profile for synchronising minimal between the Operator Client and the Master Controller
Custom	Customizable synchronisation between the Operator Client and the Master Controller
ConfigManager	Synchronisation profile for the Configuration Manager
FS	Synchronisation profile for the Forecasting Shell Server

From version 2010.01 onwards, it is possible to get an overview of the active users. This overview is available in both the Operator Client and the Admin Interface. In order to make this functionality work, the file synchProfiles.xml has to be configured properly in each of the Operator Clients. Before version 2010.01, the file synchProfiles.xml would typically contain several profiles containing the following snippet:

```

<activity id="Activity.In.FewsSessions">
  <channelId>Channel.In.FewsSessions</channelId>
  <schedule>
    <single/>
  </schedule>
  <timeOut>10000</timeOut>
</activity>

```

From version 2010.01 onwards, it is recommended to replace this snippet for the profiles 'Full', 'Minimal' and 'Custom' (not for ConfigManager) by the following:

```
<activity id="Activity.In.FewsSessions">
  <channelId>Channel.In.FewsSessions</channelId>
  <schedule>
    <continuous>
      <period unit="minute" multiplier="3" divider="1"/>
      <priority>low</priority>
    </continuous>
  </schedule>
  <timeOut>10000</timeOut>
</activity>
```

This defines that the information needed for these overviews is synchronized every three minutes.

See also [User Administration - Active Users](#)

## synchChannels

*To be completed*

## Working with pool of localDataStores (under Citrix)

A standard installation will have to be adjusted to work well in a Citrix environment. It is possible to keep your own localDatastore folder, which is then stored on your personal home directory on the Citrix server or the localDatastore can be stored centrally on the hard drive of the citrix server, in what we call a pool directory. This pool can currently exist of a maximum of 11 users at the same time (hard coded). In this case 11 localdatastore folders are created in de pooldirectory.

The following adjustments will have to be made:

1. before 2014.02 the jpif file is used together with the executable to start up the application, the last line will have to be replaced with

```
$USER_HOME$\Local Settings\Application Data\<name of application root directory>_OC
```

This is needed to make sure FEWS is working in multi-user mode, with caching in the profile directory, where \$USER\_HOME\$ is a reference to the profile of the user. At the same time you can also make use of the localDataStorePoolDir and therefore two more adjustments will have to be made.

Since 2014.02, an ini file is used instead of the jpif file. The path to the region home folder is specified in the ini-file as *arg. 1* as follows:

```
arg.1=$USER_HOME$\Local Settings\Application Data\<name of application root directory>_OC
```

2. The file \<name of application root directory>\global.properties will have to be renamed to citrix\_global.properties, to make sure you have a separate file for Citrix users.

3. in the file \<name of application root directory>\citrix\_global.properties the following line will have to be added: localDataStorePoolDir=[C:\FewsLocalDataStores]

with an extra line after this line. Please also be aware that the localDataStorePoolDir is case sensitive. Obviously you can determine yourselves where this pooldir is located.

## global.properties

The global.properties file is used to set some (Delft-FEWS) software properties at startup. See for more details, the [global.properties file](#) page.

## User\_Settings.ini

The user\_settings.ini file is a root configuration file which stores personal preferences of the user (SA and OC) like the available (un)docked windows, GUI colors etc.