02 Launching FEWS

How to setup the launch of Delft-FEWS

Launching Delft-FEWS 2017.01

A typical FEWS installation has the following subdirectories in the FEWS root directory:

1. bin: The FEWS binaries, scripts and native code.
2. jre: The java runtime
3. RegionHome: The region home with the FEWS configuration

Windows 32 and 64

Since 2017.01 FEWS is launched from the RegionHome directory. In the bin/launcher directory there are startup scripts available for all supported platforms (windows32, windows64, linux). Depending on a 64 or 32 bit environment copy the appropriate scripts to the RegionHome, for example Delft-FEWS_x64.ini and Delft-FEWS_x64.exe. Running Delft-FEWS_x64.exe will launch FEWS. The .exe can be renamed. This is convenient when having multiple FEWS system so you see the difference in the Windows Task Manager. The .ini file should always have same name as the .exe.

Linux

On linux the fews.sh script can be copied from the bin/launcher/linux directory to the FEWS root directory. Make sure the fews.sh script has executable permissions set. To run FEWS from the root directory run: ./fews.sh RegionHome. The fews.sh script also specifies the java heap size that is used by FEWS. To increase the size, this script should be tuned.

Sample Delft-FEWS_x64.ini

```
main.class=nl.wldelft.fews.gui.explorer.Application
classpath.1=../bin/*.jar
vm.location=../jre/bin/server/jvm.dll
#start up heap memory size
vmarg.1=-Xms512m
#max heap memory size, limited to 1024m for windows 32 bits
vmarg.2=-Xmx1024m
splash.image=fews-splash.jpg
```

Since 2014.02 configure a ini file

Since 2014.02 a Delft-FEWS.exe, Delft-FEWSc.exe and Delft-FEWS.ini are provided in the Delft-FEWS bin dir. Delft-FEWSc.exe is a console version of Delft-FEWS.exe and is only recommended when a console is required. The Delft-FEWS.exe can be renamed to a more suitable executable name, e.g. MyRegion_SA.exe (Stand alone), MyRegion_OC.exe (Operator Client), MyRegion_DDA.exe (Direct Database Access) or MyRegion_CM.exe (ConfigManager) depending on your needs. The executable needs to have a file with the same base name with the .ini extension, e.g. respectively MyRegion_SA.ini, MyRegion_OC.ini, MyRegion_DDA.ini or MyRegion_CM.ini. You can use the sample Delft-FEWS.ini provided in the bin directory as a starting point. Below you will find a description of the entries that can be specified in the .ini file. For specifying where to find the region home directory you will need to specify *arg.1*=<region_home>.

The two executables are copies of java launchers provided by winrun4j. (see http://winrun4j.sourceforge.net)

Sample Delft-FEWS.ini

```
#Delft-FEWS ini file
main.class=nl.wldelft.fews.gui.explorer.Application
classpath.1=*.jar

#Java Runtime jvm.dll location
vm.location=../jre/bin/client/jvm.dll
vmarg.1=-Xms512m
vmarg.2=-Xmx1024m

#location of the bin dir
working.directory=.

#region_home directory - <region_home> to be replaced with actual region home directory
arg.1=../<region_home>

#optional splash file - <region_home> to be replaced with actual region home directory
#splash.image=../<region_home>/splash_screen.jpg

#optional debug log - enable these #log and #log.overwrite for obtaining debug logging when the application
fails to start, <region_home> to be replaced with actual region home directory
#log=../<region_home>/startup_debug_log.txt
#log.overwrite=true
```

Explanation of .ini entries

vm.location=../jre/bin/client/jvm.dll	The location of the jvm.dll (from the Java JRE). On 64 bit JRE's the path is ../jre/bin/server/jvm.dll	
vmarg.<index>=-Xms512m	Initial amount of memory available for FEWS	
vmarg.<index>=-Xmx1024m	Amount of memory available for FEWS	
vmarg.<index>=-Dsun.java2d.noddraw=true	turns on/off all usage of Direct3D	
vmarg.<index>=-Djava.net.useSystemProxies=true	Use the LAN system proxy settings (default FALSE)	
vmarg.<index>=-DautoRollingBarrel=false	switch on/off the rolling barrel (default TRUE)	
classpath.1=*.jar	all the jars in the Delft-FEWS bin dir forming the classpath	
main.class=nl.wldelft.fews.gui.explorer.Application	Main class. - for the config manager use *main.class=nl.wldelft.fews.configmanagement.ConfigManagementApplication* - for the launcher use *main.class=nl.wldelft.fews.launcher.Launcher*	
working.directory=.	Location of the Delft-FEWS bin dir	
arg.1=../Anglian_SA	Path to the folder containing the configuration. Note that when the tag $USER_HOME$ is part of the foldername, it will be resolved by Delft-FEWS to the user.home directory.	

Notice that JAVA system properties should be defined after the **-D** keyword. JVM arguments are all specified with an integer index in ascending order without gaps, e.g.

vmarg.1=-XX:MaxPermSize=128m

vmarg.2=-Dsun.java2d.noddraw=true

vmarg.3=-DautoRollingBarrel=false

Before 2014.02 configure a jpif file.

jpif files are no longer supported from 2014.02 onwards. Please use the executable with the .ini file (Delft-FEWS.exe, Delft-FEWSc.exe, Delft-FEWS.ini).

In the bin from before 2014.02 there will also be a file of the same name with the extension .jpif (e.g. Your_region.jpif). Since the executable is generic (except for the name) this file contains all the information required to launch your application.

..\jre	The location of the JRE folder
-mx512m	Amount of memory available for FEWS
-cp	
$JARS_PATH$	
nl.wldelft.fews.gui.explorer.Application	Application type*
Anglian_SA	Name of folder containing the configuration

*If wanting to use the config manager use line: nl.wldelft.fews.configmanagement.ConfigManagementApplication or for the launcher use nl.wldelft.fews.launcher.Launcher

It is possible to add JAVA runtime options the .jpif file.

..\jre	JRE Folder
-Xms768m	Set initial java heap space
-Xmx1024m	Set maximum java heap space
-XX:MaxPermSize=128m	Set PermSize (default 64m, which is sometimes too low and results into PermSize errors) (not used in Java 8 and higher)
-Dsun.java2d.noddraw=true	turns on/off all usage of Direct3D
-Djava.net.useSystemProxies=true	Use the LAN system proxy settings (default FALSE)
-DautoRollingBarrel=false	switch on/off the rolling barrel (default TRUE) in the OC
-cp	
$JARS_PATH$	
nl.wldelft.fews.gui.explorer.Application	
$USER_HOME$\Application Data\FEWS\Anglian_SA	Folder name of configuration, of which a working copy will be copied and used in the defined directory. The base directory should always be at the same level a JRE and BIN

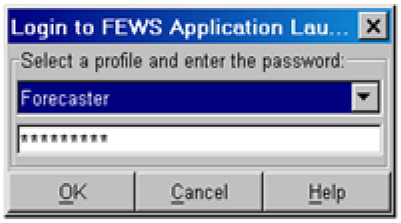
Notice that JAVA system properties should be defined after the **-D** keyword.

How to launch Delft-FEWS using the launch menu

Organisations which have multiple instances of Delft-FEWS (for example multiple regions or online/shadow type systems) may wish to use the Delft-FEWS launcher.

You must create a folder in the root directory (same level as the bin and jre) which will contain the launcher configuration files (called for example FEWSLauncher). You will need to have an executable and ini (before 2014.02 jpif) in the bin directory - you start the launcher by double clicking on the executable. But first we need to set up the launcher config files...

Firstly you can configure the password protected level of access required. Please note that this is not a highly secure method of password protection but is meant simply to restrict access to those who require it.

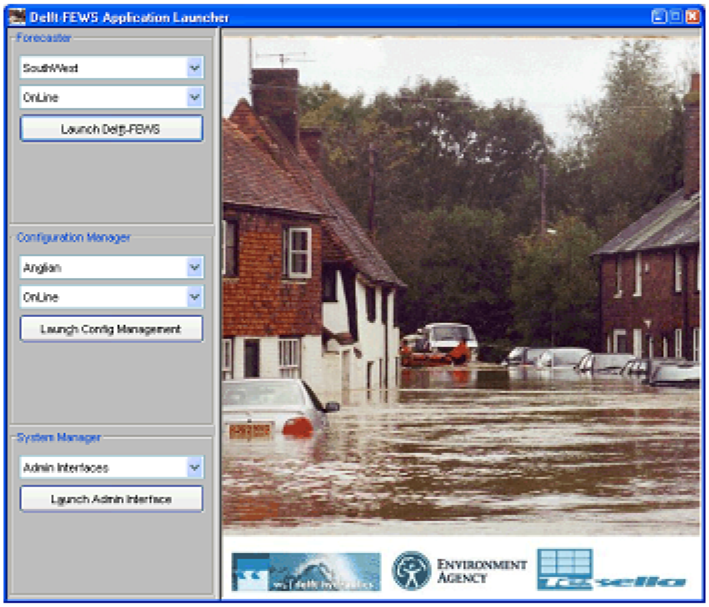

Login to FEWS Application Lau...
Select a profile and enter the password:
OK
Cancel
Help

Forecaster

****

This is done using the security.xml file (details below). The passwords are contained in the binary security.sec file.

Once you have entered the correct password you will be shown the appropriate screen from which you can choose the FEWS appplication you wish to launch.



SouthWest

OnLine

Anglian

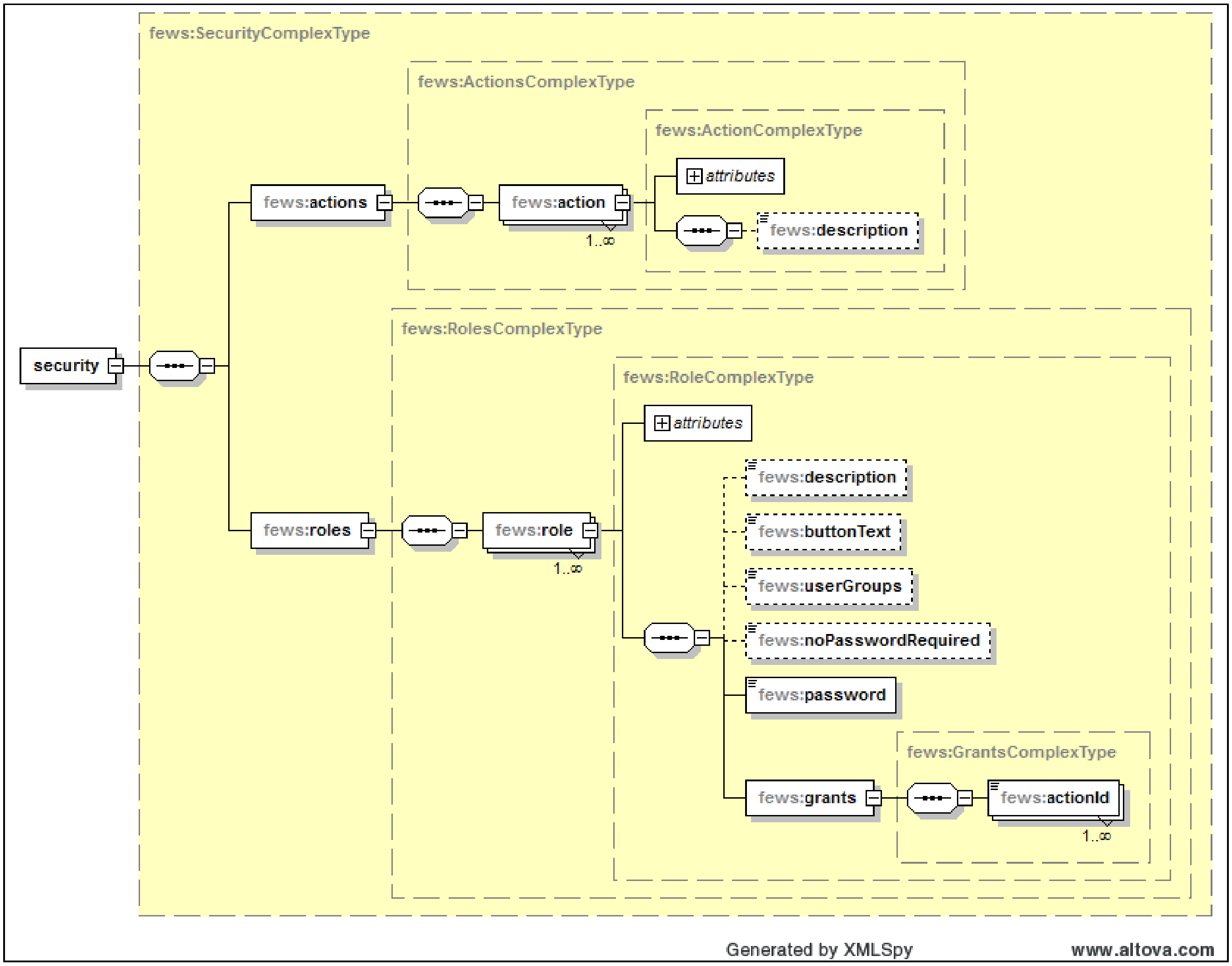
OnLine

Admin Interfaces

This is configured using the launcher.xml (details below). You can also display your organisations logo or picture of your choice by adding an image in the FEWSLauncher directory called LauncherImage.jpg of size 455 x 540 Pixels.

security.xml

The security.xml file needs to follow the diagram in the following schema.



This security.xml contains the actions and roles required. The actions are linked directly to the launcher.xml. The 'role' describes which users are have access to which actions. For example a forecaster might have access to the explorer only, while a system administrator may have access to the admin interface and configuration management interfaces. For an example file click here. This file links actionIds to user roles.

> ⚠ **UserGroups**
>
> The launcher can be used to access a client as member of the configured userGroup. Notice that this does not overrule the permission of the Windows user account. Permissions therefore can only be extended.

The following actionIds are allowed:

- ViewReports
- LaunchFewsClient,
- LaunchConfigManager,
- LaunchAdminInterface,
- Upload OnLine

In the example, the launcher is setup like the following:

- The role of Forecaster has the privileges to run the fews client (LaunchFewsClient).
- The role of ConfigManager is allowed to run fews, to run the configmanager, and upload files (LaunchFewsClient, LaunchConfigManager, Upload OnLine).
- The SystemManager is allowed what the configmanager can do as well as login to the admin interface (LaunchFewsClient, LaunchConfigManager, LaunchAdminInterface, Upload OnLine).

Howto create a security.sec file from a security.xml is specified in the privileged section, see https://publicwiki.deltares.nl/display/FEWSDOC/EncryptionTool.

security.xml can also be configured to skip the login window. For this to happen, configure as follows:

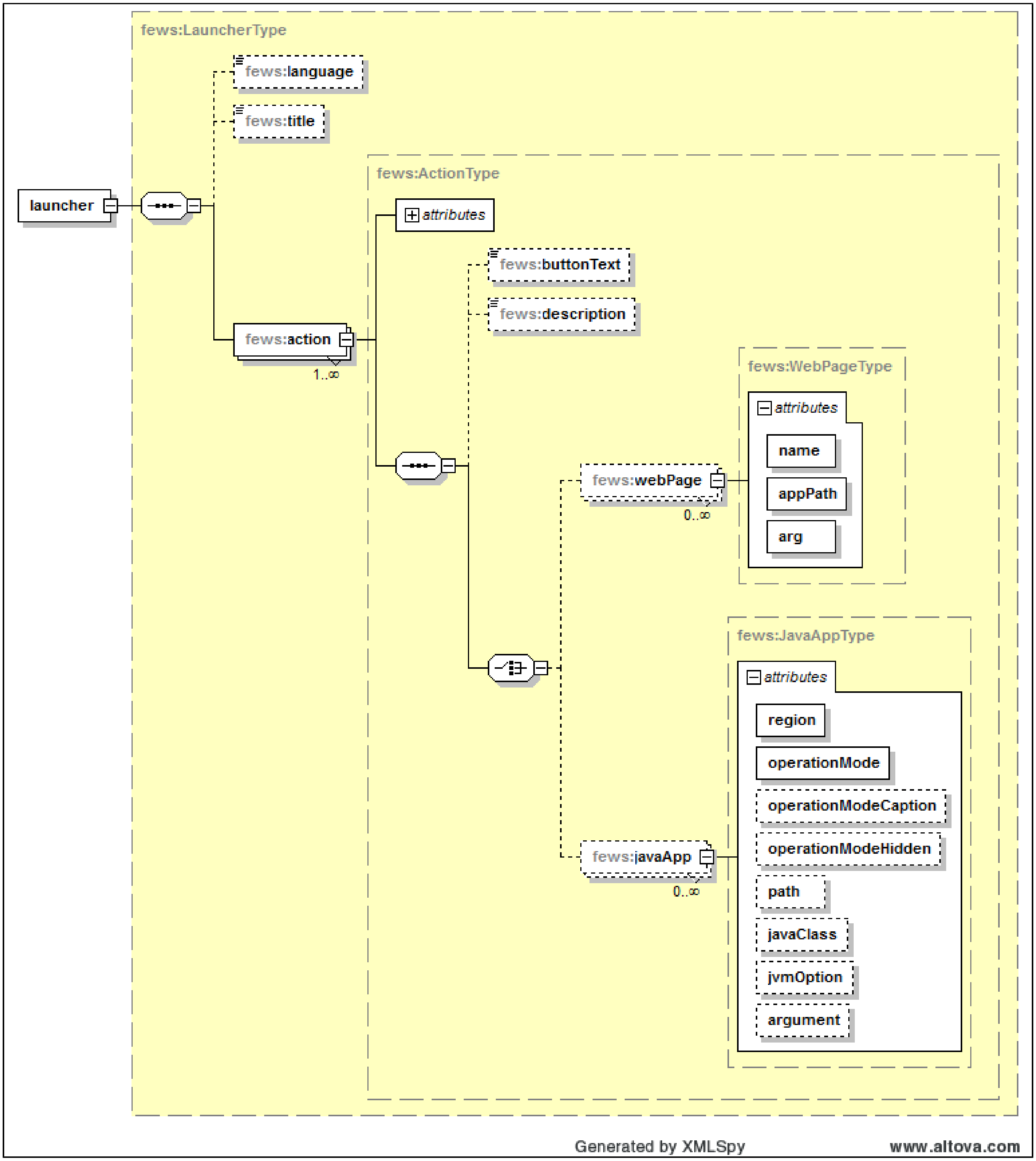
- Specify only 1 role
- The "password" element should be empty
- The "noPasswordRequired" element should be set to "true"

launcher.xml

This file contains the actions which are accessed through the launcher. The id links with the id given in the security.xml. You can see from the schema that the action can link to a web page (for example the admin interface) or to a java application (fews explorer or config manager). You will notice similarities between the attributes of the JavaAppType and those found in the .ini file in the bin directory. An example file can be seen here

The jvm Option gives you the chance to tweak the heap size used by the java virtual machine. Use -Xms for the initial java heap size (e.g. -Xms256m), -Xmx for the maximum heap size (e.g. -Xms1G). The syntax is then as follows:

```
jvmOption="-Xmx1G -Xms256m"
```



If the javaApps defined only use one operation mode (e.g. Stand Alone), it is possible to hide the operation mode dropdown box in the launcher window by setting the "operationModeHidden" attribute to true.