

03 Testing workflows with the WorkflowTestRunner in SA mode

WorkflowTestRunner

Workflows can automatically be tested on stand alone systems by using the **WorkflowTestRunner program**. The WorkflowTestRunner will start the given workflow either in the running region or start the region to run the workflow. The WorkflowTestRunner makes use of a socket interface started by the stand alone system.



Originally developed for testing purposes, the WorkflowTestRunner to run workflows automatically in a Stand Alone environment is now also used in "live" systems. This use has been proven to be performing well in practice and is now fully supported by Deltares. A full description of how to set this up, including example files, will follow shortly. See also [How to configure a workflow test run](#). [Running workflows from the command-line](#)

Explorer.xml

In order to enable automatic workflow execution on a region, one has to configure the region to listen for workflow requests on a socket interface. The socket interface can be configured in the piServicePortRange tag of the *Explorer.xml* file. Next example shows a configured pi socket for port 8432.

```
<piServicePortRange start="8432" end="8432"/>
```

A port number can only be configured once per operating system. Use different port numbers when running multiple regions on one machine. Note that the start and end attribute of the piServicePortRange are set on the same port number to make the port number a fixed port number.

Before 2014.02 Run an automatic workflow from the command line using JPIF config

Workflow runs can be started from the command line. The easiest way to accomplish this is using a jpiif configuration. In the bin directory configure the 'Region.exe' and 'Region.jpiif' files. Configure the JPIF as follows

"Region.jpiif"

```
..\jre
-mx256m
-cp
$JARS_PATH$
*nl.wldelft.fews.system.workflowrunnertest.WorkflowRunner*
<Region name>
<Workflow Id>
<ip service port nr>
\[optional system time in format: yyyy-mm-dd hh:mm:ss\]
```

Where the bold rows are WorkflowTestRunner specific. Here is an example of a valid jpiif configuration:

```
..\jre
-mx256m
-Dlog.file.path=d:\fews_wis\PeelMaasVallei\log.txt
-cp
$JARS_PATH$
nl.wldelft.fews.system.workflowrunnertest.WorkflowRunner
PeelMaasVallei
Importeer_OMC_Data
8432
2007-08-08 08:23:00
```

After starting the jpiif configuration either one of following situations can exist when the port numbers are correctly configured:

Since 2014.02, Run an automatic workflow from the command line using ini config

```

#Delft-FEWS ini file
main.class=nl.wldelft.fews.system.workflowrunnertest.WorkflowRunner
classpath.1=*.jar

#Java Runtime jvm.dll location
vm.location=../jre/bin/client/jvm.dll
vmarg.1=-Xmx256m

#location of the bin dir
working.directory=.

#region_home directory - <region_home> to be replaced with actual region home directory
arg.1=../<region_home>
arg.2=Importeer_OMC_Data
arg.3=8432
arg.4=2007-08-08 08:23:00

#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

```

case	result
The stand alone region is running and listening on the correct port number.	In in this case one should see that the workflow is executed in the system log of the explorer gui.
The stand alone region is not running.	The region will be started up. The explorer gui appears and one should see that the workflow is executed in the system log of the explorer gui.