Custom Modules – Import Custom Module

Overview

This document will describe how to package your custom 'Module' and import into the PilotFish Console.
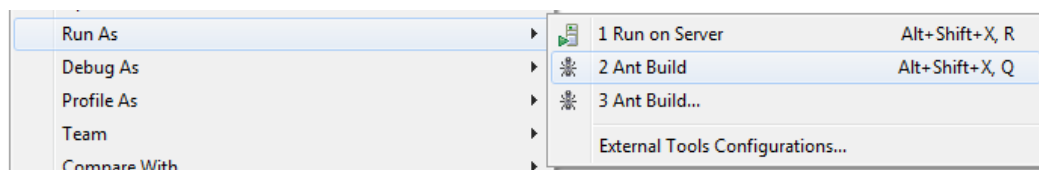
Build

The 'Module' project contains build.xml and build.properties files that will enable you to quickly package your custom module.

Open the build.properties file and make sure the "runtime.dir" property is correctly pointing at your PilotFish runtime:

    runtime.dir=/Program Files (x86)/PilotFish/eiConsole/runtime

Now right-click the build.xml file and select 'Run As' > 'Ant Build':

The logs should look like this:

```
Buildfile: C:\code\substrate\eclipse\module\build.xml
clean:
    [delete] Deleting directory C:\code\substrate\eclipse\module\build
    [delete] Deleting directory C:\code\substrate\eclipse\module\dist
init:
    [mkdir] Created dir: C:\code\substrate\eclipse\module\build\module\bin
    [mkdir] Created dir: C:\code\substrate\eclipse\module\build\test\bin
    [mkdir] Created dir: C:\code\substrate\eclipse\module\dist
compile:
    [javac] Compiling 1 source file to C:\code\substrate\eclipse\module\build\module\bin
    [javac] Compiling 1 source file to C:\code\substrate\eclipse\module\build\test\bin
archive:
      [jar] Building jar: C:\code\substrate\eclipse\module\dist\module.jar
      [jar] Building jar: C:\code\substrate\eclipse\module\dist\module_src.jar
BUILD SUCCESSFUL
Total time: 1 second
```
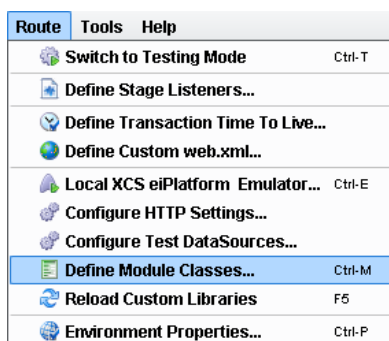
The result of the build is two JAR files; one containing source the other will be imported into the PilotFish Console.
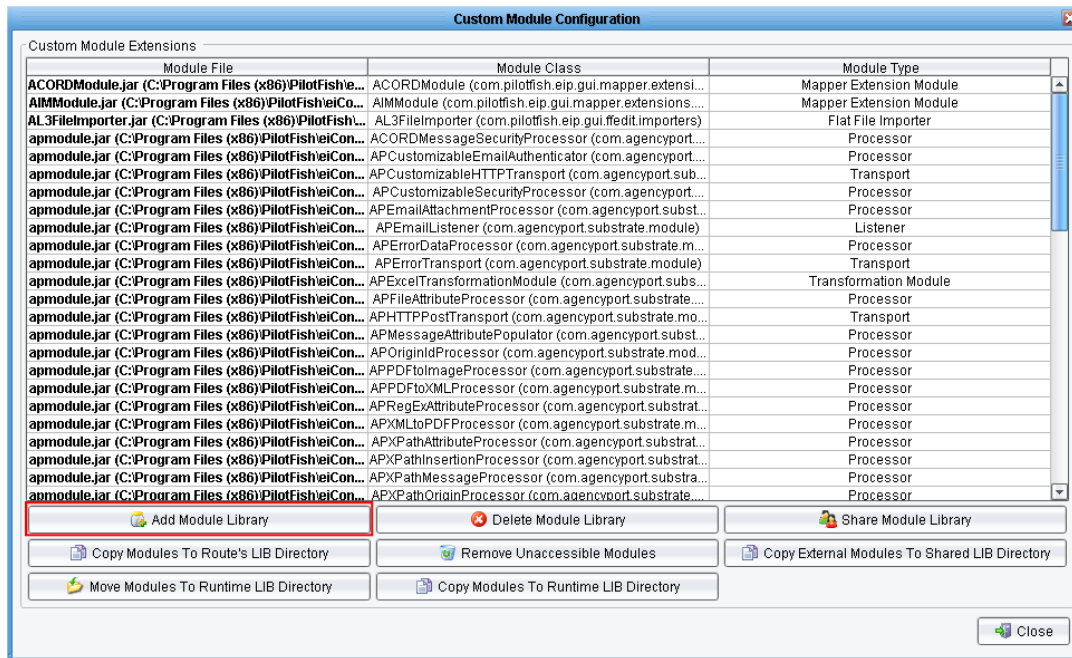
Import

If not already running, launch the PilotFish Console application.
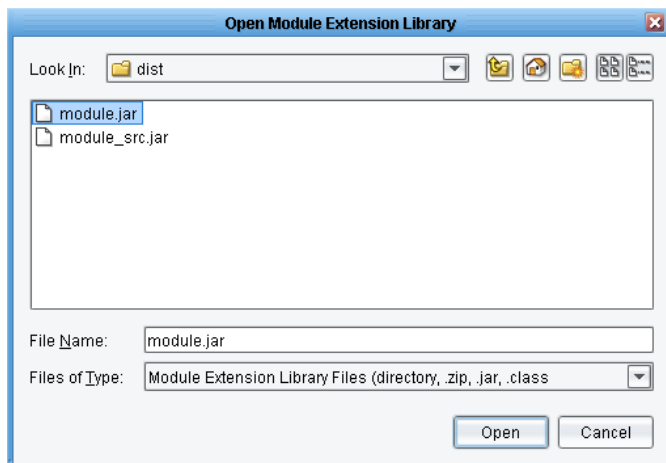
Once the application has started open one of the Routes.

From the Route Editor screen, select 'Route' > 'Define Module Classes...':

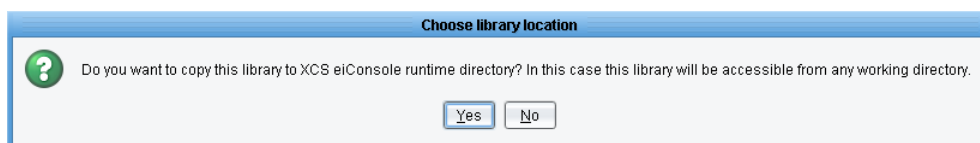| Route	Tools	Help	
	Switch to Testing Mode		Ctrl-T
	Define Stage Listeners...		
	Define Transaction Time To Live...		
	Define Custom web.xml...		
	Local XCS eiPlatform Emulator...		Ctrl-E
	Configure HTTP Settings...		
	Configure Test DataSources...		
	Define Module Classes...		Ctrl-M
	Reload Custom Libraries		F5
	Environment Properties...		Ctrl-P

From the 'Custom Module Configuration' pop-up, click 'Add Module Library' button:

Navigate to your newly created module.jar and click 'Open' button:

If prompted to 'Choose library location', click the 'Yes' button:

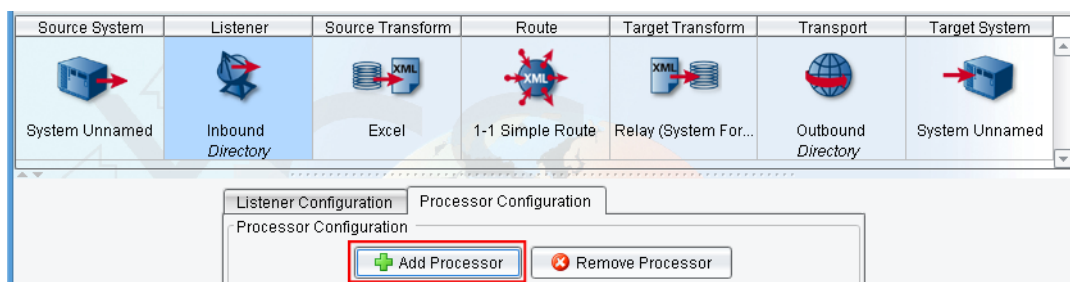You should now see your new module in the list:

You can now close the pop-up.

If you do not see your module listed you may need to close and reopen the PilotFish Console application.
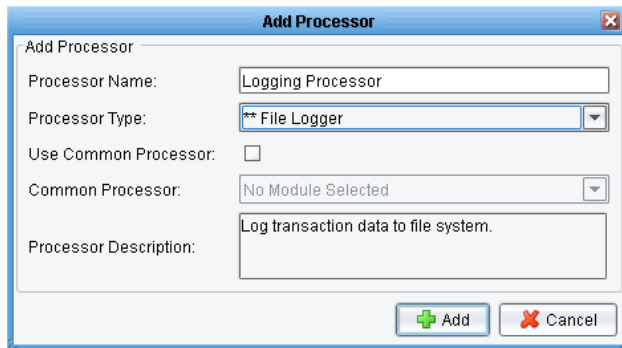
Add

You are now ready to add your custom processor to your route.
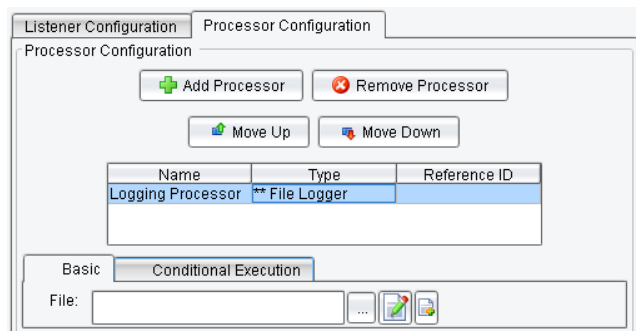
To do this; select the 'Processor Configuration' tab from the 'Listener' stage, then click 'Add Processor' button:

We will name our new processor "Logging Processor" and select "** File Logger" from the list:
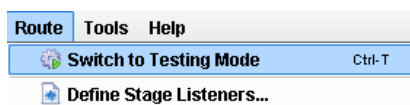
We can now add this processor and configure it:

You can either browse using the "…" button or enter the file path manually.

Testing

Now that you have imported and configured your new module you are ready to test it in the context of a route.

To do this switch to 'Route Testing' mode by selecting 'Route' > 'Switch to Testing Mode':

If prompted to 'Save Route Changes' please click the 'Yes' button.

Once in 'Test Mode' select your new processor:

We will start the test at this stage and end it right after:

You are now ready to click the 'Execute Test' button.

When prompted to 'Select File', pick one of the samples used in a previous exercise.

The result should be a green bar with 'Success' status:

Check the file path entered earlier and you should have a new file created by your custom module!