

Project Setup

[Connect5]

Table of Contents

[Overview 2](#_Toc369867289)

[Web Project (required) 2](#_Toc369867290)

[Integration Kits (optional) 2](#_Toc369867291)

[Module Project (optional) 2](#_Toc369867292)

[Outcome 3](#_Toc369867293)

# Overview

This document outlines the steps needed set up a deployable (Connect5) web project, install Integration Kits and a java project to build & debug custom modules (Processor/Listener/Transport).

We can debug modules by deploying them to the server, but it becomes cumbersome when running WebLogic or WebSphere. It’s recommend that developers create and test modules using the eiConsole Test simulator/Test route feature before deploying it to servlet container.

# Web Project (required)

Download the substrate WAR from the nightly folder and create a web project by importing the war file: [//corp-fs01/shared/00 DOCUMENTATION/Substrate/Build Artifacts/](//corp-fs01/shared/00%20DOCUMENTATION/Substrate/Build%20Artifacts/)

There is a sample Ant build file and application.xml (in case you are building an EAR file) provided. This can be found under [//corp-fs01/shared/00 DOCUMENTATION/Substrate/Documentation/Project Setup/platform](//corp-fs01/shared/00%20DOCUMENTATION/Substrate/Documentation/Project%20Setup/platform).

The build.xml expects the web root to be “Web Root”, source folder to be “src”. It also requires servlet-api.jar.

Connect5 requires a system property called “env” to load the appropriate property files. The deployment descriptor can be modified so that the initialization servlet can look for a different system property.

Connect5 requires a data source called “substrate” by default – The DDL required to create tables can be found in the nightly build folder.

# Integration Kits (optional)

Download the appropriate Integration Kits from the DevZone or Shared Folder.

Extract the zipped C5 Interfaces to the /WEB-INF/interface-root/interfaces folder.

Add the appropriate properties to /WEB-INF/properties/<env>/interface.properties.

# Module Project (optional)

The primary purpose of this project is to create and debug custom java modules.

This project depends on the libraries that are in the PilotFish eiConsole installation directory, so PilotFish eiConsole should be installed before creating this project. If you haven’t installed it yet; run the appropriate DMG or EXE files located here; <//corp-fs01/software/Pilotfish>

Create a java project within eclipse and then add a user library for the jars that are in /eiConsole/runtime and /eiConsole/runtime/lib folders.

[//corp-fs01/shared/00 DOCUMENTATION/Substrate/Documentation/Project Setup/module](//corp-fs01/shared/00%20DOCUMENTATION/Substrate/Documentation/Project%20Setup/module) has sample .classpath and pilotfish user library. With appropriate path changes, the user library can be imported in to the workspace and used.

**Steps to manually create user library:**

When creating the user library, please make sure to use “i4jruntime.jar” from eiConsole/.install4j directory.

* Add the user library to the project build path.
* Add “apmodule.jar” & “apsubstrate.jar” to the project build path (distributed separately here: [//corp-fs01/shared/00 DOCUMENTATION/Substrate/Build Artifacts](//corp-fs01/shared/00%20DOCUMENTATION/Substrate/Build%20Artifacts)).

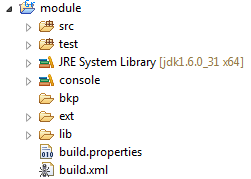
Once the project set up is complete, you can run eiConsole from eclipse by executing the “com.PilotFish.eip.gui.console.Console” main class.

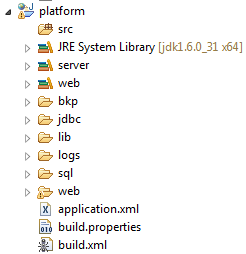
You can now set breakpoints in your custom modules and hit them when running eiConsole in Test simulator/Test route mode.

If you make any changes to the contents of “environment-settings.conf” (found in the root folder of your route configuration) must be ported to “interface.properties” (found in the WEB-INF folder of your web project).

Outcome

Your workspace setup is now complete and should looks something like the “Package Explorer” view shown below:





Below is a screenshot of the ‘web-inf’ folder and its contents, pay special attention to the ‘interface-root’ and ‘properties’ folders:

