

Connect5 Deployment Configuration

Table of Contents

[Overview 3](#_Toc371675064)

[Build Artifacts 3](#_Toc371675065)

[Deployment Diagram 3](#_Toc371675066)

[Properties 4](#_Toc371675067)

[framework.properties 5](#_Toc371675068)

[pilotfish.properties 5](#_Toc371675069)

[logging.properties 6](#_Toc371675070)

[interface.properties 7](#_Toc371675071)

[message.properties 7](#_Toc371675072)

[upload.properties 7](#_Toc371675073)

[Database 7](#_Toc371675074)

# Overview

This document outlines the artifacts that get produced during the Connect5 product build and how those artifacts should be used to setup projects within eclipse environment by the project teams.

# Build Artifacts

The following are the build artifacts that produced.

* apmodule.jar & source
  + These are the modules developed by Agencyport. Modules are descendent of EIPModule interface.
* apsubstrate.jar & source
  + These are java classes developed by Agencyport used by modules.
* Connect5.war & Connect5.ear
  + Deployable unit in a servlet container – referred to as ‘Platform’
* applatform.jar & source
  + Has initialization servlet and other java classes used by Connect5.war/Connect5.ear
* sql
  + SQL files to create tables used by platform authentication module.
* eip-root
  + Working directory that contains routes, formats, XSLT. These artifacts should be used as samples while devising solutions. At this time it’s not advised to reuse them as is. The Connect5.war/Connect5.ear file has this.

## Deployment Diagram



## Properties

Properties are grouped into 6 files they are:

* **framework.properties**
* **pilotfish.properties**
* **logging.properties**
* **interface.properties**
* **message.properties**
* **upload.properties**

### framework.properties

The **framework.properties** file is specific to an environment. The application loads the appropriate properties file from the following location **${context\_path}/WEB-INF/properties/<env>/framework.properties**. The <env> is the obtained from the ‘env’ system JVM property.

The major properties control database, logging configuration and additional properties to load.

|  |  |
| --- | --- |
| **Name** | **Description** |
| database\_agent\_class\_name | Class name of the AP Database Agent |
| datasource | JNDI URL of datasource |
| application.logging\_config\_file | The location of logging property file. |
| additional\_properties\_to\_load | The additional properties to load at bootstrap. |

Below are the default values:

**database\_agent\_class\_name=**com.agencyport.database.MySQLDatabaseAgent

**datasource=**java:comp/env/applatform

**application.logging\_config\_file**=${context\_path}/WEB-INF/properties/<env>/logging.properties

**additional\_properties\_to\_load**=${context\_path}/WEB-INF/properties/<env>/pilotfish.properties;\

${context\_path}/WEB-INF/properties/<env>/message.properties;\

${context\_path}/WEB-INF/properties/<env>/upload.properties;\

${context\_path}/WEB-INF/properties/<env>/interface.properties;

### pilotfish.properties

The **pilotfish.properties** file configures the PilotFish framework’s logging, auditing, threading, etc…

The most important property is the location for interface root (eip-root) and relative path to **interface properties**.

|  |  |
| --- | --- |
| **Name** | **Description** |
| com.pilotfish.eip.configDirectory | The location of eip-root directory. |
| com.pilotfish.eip.environmentSettings.file | The location of interface setting file. The path is relative to the eip-root directory. By default it is set to use the same framework property file. |

Below are the default values:

**com.pilotfish.eip.configDirectory**=${context\_path}/WEB-INF/interface-root/

**com.pilotfish.eip.environmentSettings.file**=/../properties/<env>/interface.properties

|  |  |
| --- | --- |
| **Name** | **Description** |
| com.pilotfish.eip.enableStageReporting | When set to true, stage processing time is recorded to database. |
| com.pilotfish.eip.enableTransactionReporting | When set to true, key transaction information is recorded to the database. |
| com.pilotfish.eip.enableParamReporting | When set to true, configured report parameters are recorded to the database. |
| com.pilotfish.eip.enableMessageLogging | When set to true, message data is written to the file system before and after exiting a route. |
| com.pilotfish.eip.messageLoggingDirectory | The location where message data will be logged. |
| com.pilotfish.eip.enableAttributeLogging | When set to true, message attributes are written to the file system before and after exiting a route. |
| com.pilotfish.eip.attributeLoggingDirectory | The location where message attributes will be logged. |
| com.pilotfish.eip.enableStageLogging | When set to true, stage data is written to the file system before and after exiting a route. |
| com.pilotfish.eip.stageLoggingDirectory | The location where stage data will be logged. |

Below are the default values:

**com.pilotfish.eip.enableTransactionReporting**=true

**com.pilotfish.eip.enableStageReporting**=false

**com.pilotfish.eip.enableParamReporting**=true

**com.pilotfish.eip.enableAttributeLogging**=true

**com.pilotfish.eip.enableMessageLogging**=true

**com.pilotfish.eip.enableStageLogging**=false

**com.pilotfish.eip.messageLoggingDirectory**=${output\_dir}/messages

**com.pilotfish.eip.attributeLoggingDirectory**=${output\_dir}/attributes

**com.pilotfish.eip.stageLoggingDirectory**=${output\_dir}/stages

Note that ‘stage’ logging is turned OFF by default. This setting is useful for debugging, but should only be used locally as it is very resource intensive.

### logging.properties

Platform uses JUL for logging. Logging level /loggers/filehandlers can be configured in the logging.properties file.

Below is the sample logging file.

**com.agencyport.portal.handlers**=com.agencyport.logging.PortalFileHandler,java.util.logging.ConsoleHandler,com.agencyport.logging.syslog.SyslogHandler

**com.agencyport.portal.loggers**=com.pilotfish

**com.agencyport.logging.PortalFileHandler.pattern**=${output\_log\_dir}/framework%g.log

**com.agencyport.logging.PortalFileHandler.limit**=20000000

**com.agencyport.logging.PortalFileHandler.count**=3

**com.agencyport.logging.PortalFileHandler.formatter**=com.agencyport.logging.DefaultLogFormatter

**com.agencyport.logging.PortalFileHandler.append**=true

**com.agencyport.logging.DefaultLogFormatter.timestamp.format**=yyyy-MM-dd HH:mm:ss:SSSSS z

**com.agencyport.logging.syslog.SyslogHandler.formatter**=com.agencyport.logging.syslog.SyslogFormatter

**com.agencyport.logging.syslog.SyslogHandler.level**=SEVERE

**com.agencyport.logging.syslog.SyslogHandler.host**=127.0.0.1

**com.agencyport.logging.syslog.SyslogHandler.port**=514

**java.util.logging.ConsoleHandler.level**=SEVERE

**java.util.logging.ConsoleHandler.formatter**=com.agencyport.logging.DefaultLogFormatter

**com.pilotfish.useParentHandlers**=false

**com.pilotfish.eip.transact.level**=SEVERE

**com.pilotfish.eip.server.level**=SEVERE

**com.pilotfish.eip.cache.level**=SEVERE

**com.pilotfish.eip.extend.level**=SEVERE

**com.pilotfish.eip.admin.level**=SEVERE

**com.pilotfish.eip.config.level**=SEVERE

**com.pilotfish.eip.modules.level**=SEVERE

### interface.properties

This file stores all environment properties used in your routes and interfaces (e.g. URLs). Its design time equivalent is the **environment-settings.conf** file.

Currently these files have to be kept in synch manually, note that the contents of the **.conf** file are escaped (e.g. http\:), whereas **.properties** are not. Lastly the use of ${property} referenced variables is not allowed in the **.conf** file.

### message.properties

This file is used much like a resource bundle, without the localized content. It houses all user facing messages for easy editing.

### upload.properties

This file is used only by the File Selector Widget. See ‘Portal File Selector Widget.doc’ for more information.

# Database

Use the **reload.sql** found with the sql folder to create tables. It has SQL statements to drops and creates tables used by platform. When creating the database for the first time, remove the drop statements.

