

Page Flow Project Template

Table of contents

1 Introduction.....	2
2 Using the Page Flow Web App Template.....	2
2.1 Copy and Rename the Template Folder.....	2
2.2 Edit the build.properties File.....	2
2.3 Build the Template Web App.....	3
2.4 To Deploy and Run the Template Web App.....	3

1. Introduction

Use the Page Flow application template (located at `<BeehiveRoot>/samples/netui-blank`) as a starting point for building your own Page Flow applications.

Note:

`<BeehiveRoot>` refers to the top-level directory of your Beehive installation. A typical value for `<BeehiveRoot>` would be `C:/apache/apache-beehive-1.0.`

The template contains everything needed in a basic Page Flow application, except for the runtime and `<netui>` tag library JAR files. This topic explains how to add those JAR files and get the template up and running.

2. Using the Page Flow Web App Template

The following instruction assume that you have completed all of required and optional steps in the Beehive set up procedure at [Installation and Setup](#) (`../setup.html`) .

To use the template, follow these steps:

2.1. Copy and Rename the Template Folder

Copy the folder `<BeehiveRoot>/samples/netui-blank` to your development directory.

Below we assume that you have copied `netui-blank` into the folder **C:/myDevelopmentDirectory**.

```
C:/myDevelopmentDirectory/netui-blank
```

Rename `netui-blank` to something more appropriate to your application. Below we assume that `netui-blank` has been renamed as **myWebApplication**.

```
C:/myDevelopmentDirectory/myWebApplication
```

2.2. Edit the build.properties File

In this section you will edit the `build.properties` file--the file that sets the build-related properties for your web application.

Open the file

```
C:/myDevelopmentDirectory/myWebApplication/WEB-INF/src/build.properties
```

in a text editor.

Page Flow Project Template

Edit the `beehive.home` property so that it points to the top-level folder of your beehive installation.

Edit the `context.path` property to some value that is appropriate to your web application.

For example, if your beehive installation resides at `C:/apache/apache-beehive-1.0`, then your `build.properties` file would appear as follows.

Note: the value of `<SomeContext>` will determine the app's URL address.

```
beehive.home=C:/apache/apache-beehive-1.0

servlet-api.jar=${os.CATALINA_HOME}/common/lib/servlet-api.jar
jsp-api.jar=${os.CATALINA_HOME}/common/lib/jsp-api.jar

context.path=<SomeContext>
```

Note:

Windows users must use forwardslashes (/) not backslashes (\) in the `build.properties` file.

2.3. Build the Template Web App

To build the web app, run the following Ant command. This Ant command will delete any build artifacts in the `classes` directory, compile the web application source, and archive the result as a WAR file. The WAR file will be saved at

`C:/myDevelopmentmentDirectory/`. The name of the WAR file will be `<SomeContext>.war`.

```
ant
-f C:\myDevelopmentDirectory\myWebApplication\WEB-INF\src\build.xml
clean
build
war
```

Copy and Paste version:

```
ant -f C:\myDevelopmentDirectory\myWebApplication\WEB-INF\src\build.xml
clean build war
```

2.4. To Deploy and Run the Template Web App

Before deploying the application, ensure that Tomcat is turned on.

```
%CATALINA_HOME%\bin\startup.bat
```

To deploy the web application copy the application WAR file to Tomcat's `webapps` dir.

```
copy C:\webDevelopmentDirectory\<SomeContext>.war %CATALINA_HOME%\webapps
```

You can now try out the web app by pointing your browser at the following link.

- Run the controller file's `begin()` method: <http://localhost:8080/<SomeContext>>/

Java, J2EE, and JCP are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.

Java, J2EE, and JCP are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.

© 2004, Apache Software Foundation