



Jetspeed-2 Deploy Tools v.2.1.2

Project Documentation

Table of Contents

1 Jetspeed-2 Deploy Tools Documentation	
1.1 Jetspeed-2 Deployment Overview	1
1.2 Deploy Tools Overview	4

1.1 Jetspeed-2 Deployment Overview

Jetspeed-2 Deployment Overview

Deploying Portlets in Jetspeed-2: An End User View

Deploying custom portlets in Jetspeed-2 is simple. Portlets are very similar to servlets. They require a deployment descriptor, `portlet.xml` which goes in `WEB-INF` and need to be packaged in a war-like format. You can find quite a few example `portlet.xml` files within the Jetspeed-2 source tree. For starters take a look at the one under `/portal/src/webapp/WEB-INF`. In order to deploy a portlet, Jetspeed-2 requires the user to follow those steps:

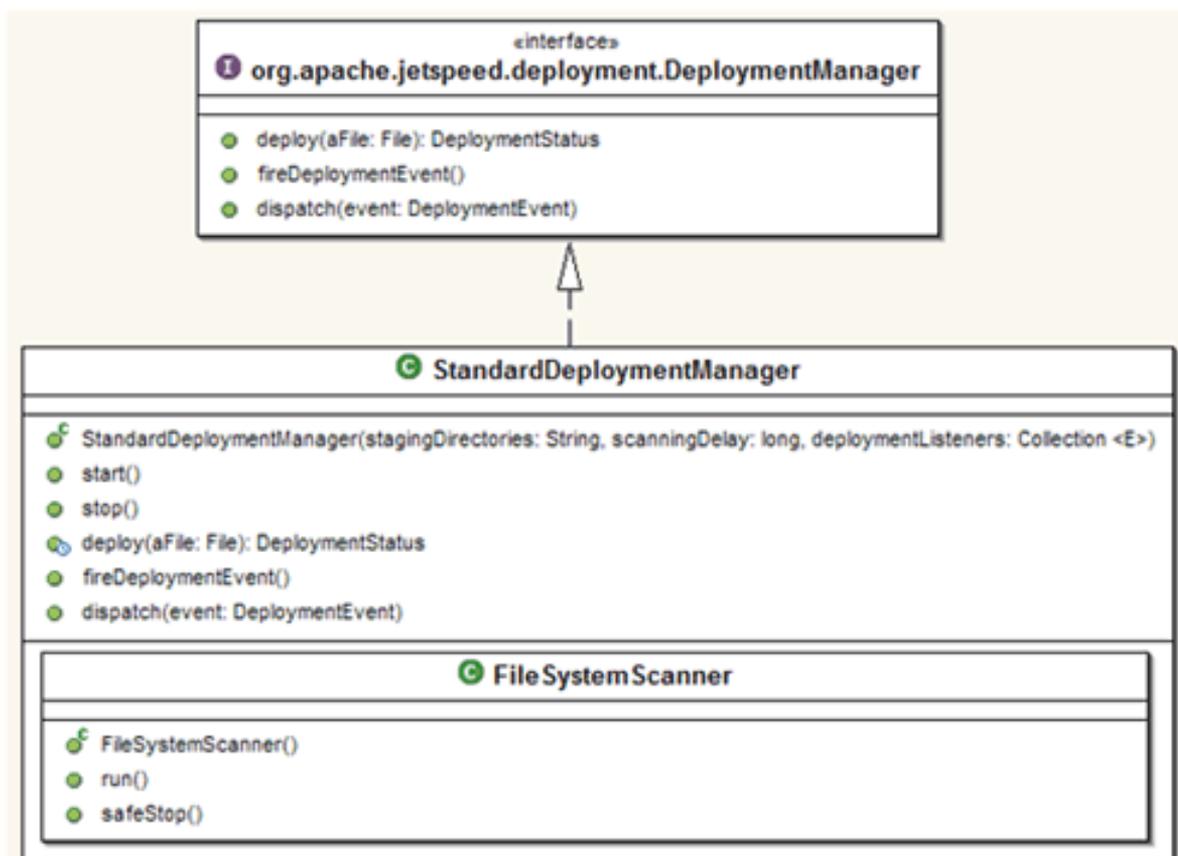
1. Build your portlet as a portlet application just as you would a web application.
2. Package your portlet application into a `.war` file.
3. Copy the `.war` file to Jetspeed's deployment directory, by default this is `WEB-INF/deploy`. Jetspeed will take care of automatically deploying the portlet into the portlet registry and will also deploy the portlet as a web application into the app server Jetspeed is deployed to.
4. The easiest way to view your portlet is to add an entry to the `default-page.psm1` under `jetspeed/WEB-INF/pages`. The id for the portlet fragment uses a unique combination of `${portlet.application.id}::${portlet.name}` where `${portlet.application.id}` is the actual name of the war file (minus the ".war") that contains your portlet app and `${portlet.name}` needs to be the value in the portlet name tags, `<portlet-name>MyPortlet</portlet-name>`. Changes to the `psml` will be picked up automatically and you should now be able to view your portlet!

Portlet Deployment: How Does it Work?

The component hierarchy below describes the assembly dependencies supporting Jetspeed-2 deployment functionality.

- The `DeployPortletAppEventListener` handles the hot deployment of portlet applications.
- The `DeployDecoratorEventListener`: handles the hot deployment of decorators. See [guide to decorators](#) for more information.

Jetspeed-2 provides a `StandardDeploymentManager`. The `StandardDeploymentManager` leverages a `FileSystemScanner` to scan for new assets to deploy. It leverages Jetspeed-2 [deploy tools](#) to prepare portlet applications prior to deployment.



1.2 Deploy Tools Overview

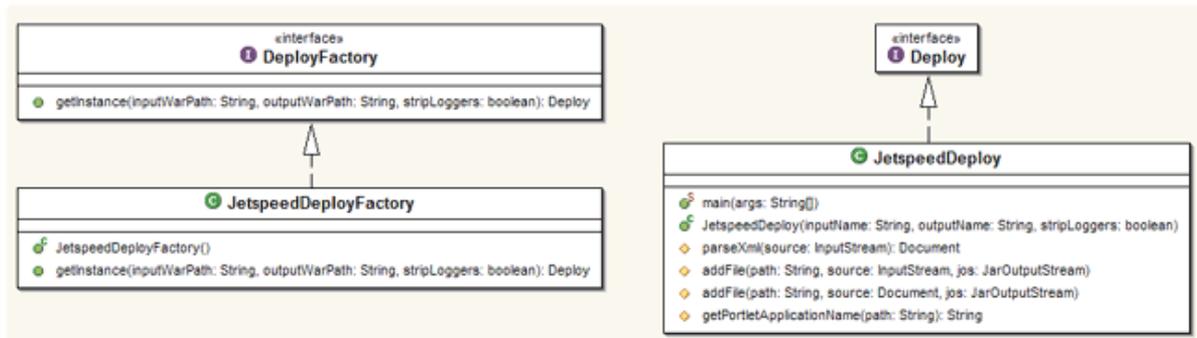
The Role of Jetspeed-2 Deploy Tools

JetspeedDeploy and the DeploymentManager

`JetspeedDeploy` prepares portlet applications for deployment within Jetspeed-2. When a new portlet deployment event is registered, the `DeployPortletAppEventListener` invokes `JetspeedDeploy` to prepare the portlet application for deployment.

```
new JetspeedDeploy(event.getPath(), toFile.getAbsolutePath(), stripLoggers);
```

`JetspeedDeploy` copies the web application archives (.war) from the input directory to the output directory and parses the `web.xml`, `portlet.xml`, and `context.xml` to ensure their compliance with the Jetspeed-2 portal engine.



`JetspeedDeploy` invokes the `JetspeedWebApplicationRewriter` to infuse the `web.xml` with the `JetspeedContainer` servlet if it does not already exist:

```
<servlet>
  <servlet-name>JetspeedContainer</servlet-name>
  <display-name>Jetspeed Container</display-name>
  <description>MVC Servlet for Jetspeed Portlet Applications</description>
<servlet-class>org.apache.jetspeed.container.JetspeedContainerServlet</servlet-class>
  <init-param>
    <param-name>contextName</param-name>
    <param-value>${portlet-application-name}</param-value>
  </init-param>
  <load-on-startup>0</load-on-startup>
</servlet>
...
<servlet-mapping>
```

```
<servlet-name>JetspeedContainer</servlet-name>
<url-pattern>/container/*</url-pattern>
</servlet-mapping>
```

In the same fashion, the `JetspeedDeploy` invokes the `JetspeedContextRewriter` to manipulate a portlet application context `.xml` file. For more information about Tomcat context `.xml`, see [tomcat's documentation](#).

JetspeedDeploy Standalone Usage

`JetspeedDeploy` can also be invoke through the command line:

```
java -jar jetspeed-deploy-tools-<version>.jar -s inputWarPath outputWarPath
```

where:

- `-s`: flag indicating whether or not to strip to loggers from the application. When the flag is present, the loggers available in the application will be removed.
- `inputWarPath`: the path of the war to process.
- `outputWarPath`: the path of the processed war.