

Documentation for Designers working with Jetspeed Decorations

by David Sean Taylor

Table of contents

1	2
2	2
3	2

1.

Decorators are defined as any static or semi-static markup surrounding a dynamically generated Fragment. Decorators are usually written with either Velocity or JSP templates. Decorators define how a single portal page is aggregated. A layout is made up of:

- One or more templates
- A template descriptor
- Images
- A Stylesheet (CSS)
- Macros

2.

Layouts are packaged in a JAR file. Supported decorators:

- Page Decorators
- Portlet Decorators
- Generic Decorators
- Localized Decorators

Operations Supported:

- Deploy
- Un-deploy
- Re-deploy

The `decorator.properties` file must be located in the root directory of the jar archive and contain the properties outlined in the next slide to be identified as a decorator by the auto deployment system. The deployer will relocate the locale specific content to satisfy J2 content locators as necessary on deploy and undeploy. The current implementation supports only 2 character language and country codes.

3.

The `decorator.properties` file must be located in the root directory of the jar archive and contain the properties outlined above to be identified as a decorator by the auto-deployment system.

A deployable decorator must have the following properties set or defaulted in the `decorator.properties` configuration:

```
id = IDENTIFIER
media.type= HTML | WML
decorates = layout | portlet | any
```

To avoid conflicts with the locale logic and runtime locators, decorators should be named with ids longer than 2 characters in length. The `media.type` property defaults to 'html'. The 'any' value for the `decorates` property is mapped to the generic deployment directory. The deployer will relocate the locale specific content to satisfy J2 content locators as necessary on deploy and undeploy. The current implementation supports only 2 character language and country codes.