

## Preface

This document is the Portlet 2.0 Bridge Specification for JavaServer™ Faces 1.2. It defines the subsystem that allows a JavaServer Faces resource to execute and satisfy a portlet request. The bridge is layered between the portlet container and the JavaServer™ Faces runtime. Its behavior and implementation depends on both. There will be a discrete bridge specification and implementation for each version combination of JavaServer Faces and Java Portlets. This version specifies the bridging function necessary to execute JavaServer™ Faces version 1.2 pages in a Java™ Portlet specification version 2.0 environment. For information on executing JavaServer™ Faces version 1.2 pages in a Java™ Portlet specification version 1.0 environment see the related specification “Portlet 1.0 Bridge Specification for JavaServer™ Faces 1.2”.

### 1.1 Portlet 1.0 Compatibility

Because the Portlet 2.0 container is fully backwards compatible with the Portlet 1.0 specification, a JSF application designed to run in a Portlet 1.0 Bridge environment will run unchanged in a Portlet 2.0 Bridge environment.

### 1.2 Additional Sources

The specification is intended to be a complete and clear explanation of the portlet bridge function. If questions remain the following may be consulted:

- A reference implementation (RI) has been made available. It is published in open source in the [PortletBridge subproject of the Apache MyFaces project](#). Though contained under the MyFaces umbrella, its use is not restricted to or by any other MyFaces implementation. It should run on any Faces 1.2 implementation and in any portlet container. This RI provides a behavioral benchmark for this specification. Where the specification leaves implementation of a particular feature open to interpretation, implementers may use the reference implementation as a model that illustrates the intention of the specification.
- A Technology Compatibility Kit (TCK) has been provided for assessing whether implementations meet the compatibility requirements of the Portlet Bridge standard. The test results have normative value for resolving questions about whether an implementation is standard.

- If further clarification is required, the working group for the Portlet Bridge under the Java Community Process should be consulted. It is the final arbiter of such issues.

Comments and feedback are welcomed, and will be used to improve future versions.

### 1.3 Who Should Read This Specification

The intended audience for this specification includes the following groups:

- **Bridge implementor:** e.g. Java™ Portlet container implementers that want to expand the reach of their environment to support portlets whose implementations are either in part, or in whole, implemented using JavaServer™ Faces.
- **Portlet developer:** e.g. The portlet author who prefers to use the MVC Java standard to author her portlets.
- **JSF developer:** e.g. JavaServer™ Faces authoring tool developers who wish to support web applications that publish portions of their Faces views as portlets or JavaServer™ Faces extension developers who want to ensure that their extension can run in a portlet (bridge) environment.

The specification is not a user's guide for developing portlets using JavaServer™ Faces and is not intended to be used as such. As it focuses on the behavioral details of the bridge, it is primarily oriented towards bridge implementors. Portlet developers or JSF developers who want quick access to the information that is most pertinent to them should consult:

#### Portlet Developer:

*Basic information:*

- [Chapter 2](#): this chapter provides a short overview of portlets, JSF, and the bridge architecture. It also introduces you to the terminology used in the specification.
- [Chapter 4](#): this chapter defines the behavior of the GenericFacesPortlet; a bridge provided portlet implementation that you will commonly subclass to build your portlet. It hides all the details of initializing, configuring, managing, and executing the bridge.
- [Chapter 3](#): this chapter defines the bridge interface and configuration which a portlet uses to initialize, configure, manage, and execute it. I.e. defines the interface and configuration details hidden to you in chapter 4 by the GenericFacesPortlet implementation.

- [Chapter 6, section 6.5](#) through subsection 6.5.2.1 plus section 6.5.2.3: these sections provide information related to portlet specific EL evaluation.

*Advanced Information:*

- [Chapter 6, section 6.8](#): JSF provides a facility allowing managedbean implementors to be notified before the bean is destroyed. This section includes important information for supporting this notification (correctly) in a portlet environment.
- [Chapter 5, section 5.1](#): this section describes how the portlet and JSF lifecycles differ and how the bridge manages state to ensure expected behavior in JSF. You will need to understand this information and how to exclude specific state from being managed by the bridge if you determine that such state management is detrimental/not needed.
- [Chapter 7](#): this chapter defines the behavior of a servlet filter that can be used with the bridge to provide correct markup ordering when your JSF view is represented by a jsp whose output comes from regular jsp markup as well has JSF components. Note: filter implementations aren't part of the formal specification because JSF is structured in such a way as to require distinct implementations per JSF implementation. If you need such a filter to correct the jsp ordering problem you will have to search for a publicly available implementation for your JSF implementation type or write one yourself.

## 1.4 API Reference

An accompanying javadoc™ includes the full specifications of classes, interfaces, and method signatures.

## 1.5 Important Dependent Java™ Specifications

As a bridge between a Java Portlet v2.0 container and a JavaServer Faces v1.2 environment, this specification depends directly on each of these specifications:

- [JavaServer™ Faces v1.2](#)
- [Java Portlet Specification v2.0](#)

These specifications are available in their respective JSR homes (JSR 252; JSR268) on <http://www.jcp.org>.

## 1.6 Dependent Java™ Versions

The portlet bridge should run on any Java Portlet 1.0 container. It should also run on a portlet container running a newer version of the Java Portlet specification. However, it is recommended that the version of the bridge be paired with both the version of the Java Portlet container and Java Server Faces runtime it is deployed with. This ensures the best implementation for the given environment.

The portlet bridge requires, as a minimum, the JSF runtime support equivalent of JSF 1.2MR1. This corresponds to the JSF 1.2\_03 version of the reference implementation. Specifically, the JSF runtime must call `ExternalContext.encodeActionURL` on the path before passing the result to `ExternalContext.redirect()`.

In addition, the bridge requires a Java 1.5 environment because this is the minimum required version for running JSF 1.2.

## 1.7 Terminology

The key words MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, SHOULD, SHOULD NOT, RECOMMENDED, MAY, and OPTIONAL in this document are to be interpreted as described in [RFC2119].

## 1.8 Providing Feedback

We welcome any and all feedback about this specification. Please e-mail your comments to [jsr-301comments@jcp.org](mailto:jsr-301comments@jcp.org). Please note that due to the volume of feedback that we receive, you will not normally receive a reply from an engineer. However, each and every comment is read, evaluated, and archived by the specification team.

## 1.9 Acknowledgments

The Portlet 2.0 Bridge Specification V1.0 is the result of the work of JSR 329 Expert Group which migrated from JSR 301: Subbu Allamaraju, Brodi Beartusk, Andy Bosch, Matt Brasier, Benjamin Bratkus, Wesley Budziwojski, Raschka Christian, Shankar Djeyassilane, Nikolai Dokovski, Ate Douma, Deepak Gothe, Ted Goddard, Wesley Hales, Jondean Haley, Roger Kitain, Danny Machak, Kito Mann, Martin Marinschek, Scott O'Bryan, Stan Silvert, Thomas Spiegl, Shashank Tiwari, Julien Viet, James Ward, and Carsten Ziegeler. Besides those above representing themselves, the following companies were represented in the expert group: BEA Systems, SAP AG, Oracle Corporation, HIPPO, Sun Microsystems, Icesoft Technologies Inc., Red Hat Middleware LLC., Adobe Systems, and S&N AG. Though the contributions of all were important, special

thanks is given to Nikolai Dokovski of SAP AG and Scott O'Bryan of Oracle Corporation who were particularly active throughout the entire specification process and provided immeasurable support and feedback.

[Previous](#)

Portlet 2.0 Bridge for JavaServerTMFaces 1.2 – November 14th, 2010

[Next](#)