Printed: Feb 20, 2002
From: http://jcp.org/jsr/detail/168.prt

Standard Format

JSRs: Java Specification Requests
Detail

JSR 168
Portlet Specification

To enable interoperability between Portlets and Portals, this specification will define a set of APIs for Portal computing addressing the areas of aggregation, personalization, presentation and security.

Specification Lead

Alejandro Abdelnur
Sun Microsystems, Inc.
Stefan Hepper
IBM

Expert Group

IBM Sun Microsystems, Inc.

I would like to join this Expert Group.

Private Expert Group homepage 

Status

JSR Review Ballot 11 Feb, 2002

Java Community Process version in use: 2.1

Please direct your comments on this JSR to jsr-168-comments@jcp.org

Original Java Specification Request (JSR)

Identification | Request | Contributions

Section 1. Identification

Submitting Member: IBM & Sun Microsystems, Inc.

Name of Contact Person: Adam Abramski & Thomas Schaeck

E-Mail Address: adam.abramski@sun.com, schaek@de.ibm.com

Telephone Number: +1 408 276 6378, +49 171 692 8407

Fax Number: +1 408 276 4283, +49 7031 16 4888

Specification Lead: Alejandro Abdelnur, Stefan Hepper

E-Mail Address: alejandro.abdelnur@sun.com, sthepper@de.ibm.com

Telephone Number: +1 408 276 5207, +49 7031 16 3445

Fax Number: +1 408 276 4283, +49 7031 16 4888

Initial Expert Group Membership:

Apache Software Foundation
BEA
IBM
Sun Microsystems

Supporting this JSR:

Accenture
Apache Software Foundation
BEA
Boeing
Borland
Bowstreet
Cap Gemini Ernst & Young
Citrix
DaimlerChrysler
Documentum
Enformia Ltd
Epicentric
Hewlett-Packard
Interwoven
Macromedia
McDonal Bradley
Oracle
Plumtree
SAP
Silverstream
Sybase
Tarantella, Inc
Vignette

Section 2: Request

2.1 Please describe the proposed Specification:

The Portlet specification will define a Portlet API that provides means for aggregating several content sources and applications front ends. It will also address how the security and personalization is handled.

Portlets are web components -like Servlets- specifically designed to be aggregated in the context of a composite page. Usually, many Portlets are invoked to in the single request of a Portal page. Each Portlet produces a fragment of markup that it s combined with the markup of other Portlets, all within the Portal page markup.

The Portlet specification will define the different components for Portal Computing, their interaction, lifecycle and semantics. These components will comprise -but they will not be

restricted to-: Portlets, Deployment descriptor, Portlet APIs. In addition, APIs for vendor extensions, APIs for security, user customization and layout management will be considered.

Also, it will define the minimum set of possible window states for a Portlet such as normal, minimized, maximized, etc.-, the valid state transitions and Portlet modes (such as view, edit, help, configure) per markup language.

This first version of the Portlet specification will concentrate in the following design goals:

- Client agnostic
- Support for multiple types of clients (multi-device)
- Simple Portlet API
- Support for Localization and Internationalization
- Hot deployment and re-deployment of Portal applications
- Declarative security (same as to the mechanism found in Servlet and EJB specs)
- Architected to support remote execution of Portlets

The Portlet specification will be based on the Servlet specification. It is envisioned that the developer API will be similar to the Servlet API.

The Portlet specification will restrict the use of functions provided by the Servlet API to a subset that makes sense for components providing fragments of a markup page.

Portlets would be able to obtain from their context -via the Portlet API- functions like access to user profile information for the current user, participation in the portal window and action event model, access to web client information, sharing of information with other Portlets and a standard way of storing and retrieving per-user/per-instance Portlet data persistently.

The API will provide a URL-rewriting mechanism for creating links to trigger actions within a Portlet without requiring knowledge on how URLs are structured in the particular web application.

Portlets would be grouped in a Portal Application by bundling them in a single WAR with a Portlet deployment descriptor file. In addition, the API will provide a mean for sharing data among Portlets of the same Portal Application.

Like the Servlet specification, the Portlet specification will allow access to Enterprise Information Systems without imposing restrictions on the type of protocols.

It is an important goal that the design of the Portlet specification would allow

implementations to support remote Portlet execution. This design would not address the transport protocol for the remote execution of Portlets, leaving to the specific Portal implementations the support for Portlet remote execution. For example, a proxy Portlet could be used to invoke a remote Portlet.

The Expert group will consider functionality such as support for, parallel execution of Portlets within a single user request, logging, security and personalization.

The Expert group will decide if the specification should include a set of specialized Portlet implementations for common tasks such as syndication (RSS), HTML scrapper, Web Services access, etc.

The Expert Group will evaluate defining a Credential mapping service to allow the Portal application to access resources in other applications not supporting the notion of distributed sessions- on behalf of user.

It is understood that the subject of this JSR is already being addressed by Open Source projects and products from different vendors. The Objective of this JSR is to create a standard for Java Portal Applications, which will help unifying a fragmented area. The expert group will ensure this specification draws appropriately from such projects and products and that it will be based on open standards.

2.2 What is the target Java platform? (i.e., desktop, server, personal, embedded, card, etc.)

A Java extension for the J2EE 1.4 platform.

2.3 What need of the Java community will be addressed by the proposed specification?

This specification will establish a standard API for creating Portlets, thus avoiding locking in Portal developers in a specific implementation and allowing Portlets developers to reach a wider audience while reducing their development efforts.

The Portlet specification is required to achieve interoperability between Portlets and Java-based Portal servers or other web applications that implement the specification. The goal is to allow Portlets to be packaged into WAR files and deployed in a standard way on any server implementing the specification.

2.4 Why isn't this need met by existing specifications?

While the Servlet/JSP specifications define an include mechanism for aggregating

Servlets and JSPs, they do not define the Desktop metaphor where this aggregation happens. Neither the Servlet/JSP specifications define the possible states and transitions of an included Servlet or JSP, or how the state of one Servlet or JSP affects the display of the other included Servlets or JSPs. In addition, The Servlet/JSP specifications do not define a personalization interface or the idea of persisting the personalization information. Furthermore, the Servlet specification does not define URL-rewriting functions to allow the creation of links and actions targeted to a specific from within the fragment of a page (Portlet markup fragment).

The Java Server Faces (JSR 127) aims to define a standard, MVC based, Web GUI framework focusing on the UI components (input fields, lists, buttons, etc.) and their event model. However, it does not address aggregation, security and personalization.

2.5 Please give a short description of the underlying technology or technologies:

The Portlet specification will be designed leveraging the following technologies: XML, JAXP, Servlet/JSP, JAAS and other J2EE technologies.

For example, a JSP tag library extension or Java Server Faces implementation could be used by a Portlet developer to render the Portlet s content. In addition, a JSP tag library extension or Java Server Faces could be used by a Portal vendor to implement the rendering of the Portal page.

For a description of the Portlet technology, refer to section 2.1.

2.6 Is there a proposed package name for the API Specification? (i.e., `javapi.something`, `org.something`, etc.)

javax.servlet.portlet.

2.7 Does the proposed specification have any dependencies on specific operating systems, CPUs, or I/O devices that you know of?

No.

2.8 Are there any security issues that cannot be addressed by the current security model?

No

2.9 Are there any internationalization or localization issues?

Yes. APIs and descriptors to support internationalization and localization are a fundamental design goal of this JSR

2.10 Are there any existing specifications that might be rendered obsolete, deprecated, or in need of revision as a result of this work?

No

2.11 Please describe the anticipated schedule for the development of this specification.

To be determined by the expert group, initial target is December 2002.

To reach this target the following schedule may be used as starting point:

Portlet API Spec community draft: 05/2002
Portlet API Spec public draft: 07/2002
Portlet API final draft: 10/2002
Reference Implementation & TCK: 12/2002

2.12 Please describe the anticipated working model for the Expert Group working on developing this specification.

We anticipate a mixture of mailing list and occasional face to face or teleconference meetings.

It is expected that both specification leaders will fully share responsibilities associated with group leadership, including group communications, decision making, and agreeing to the business terms for the RI and TCK. Exact details will be agreed early in the life of the JSR and communicated to expert group members.

The RI will be managed by IBM as an open source project at Apache and will be made available under terms similar to that used for Apache Tomcat.

The specification, RI, and TCK will be freely available for independent implementations. The TCK will be managed by Sun and will be available to independent implementors with no requirements to also license or use the RI. There will be no shared code requirements.

If this specification, or a future version of this specification, is included in a future version of a Java platform specification, this specification will remain available for use outside the platform specification, and will continue to be evolved outside the platform specification,

unless both specification leads agree otherwise.

Section 3: Contributions

3.1 Please list any existing documents, specifications, or implementations that describe the technology. Please include links to the documents if they are publicly available.

Different implementations are available today, the following list enumerates some of them:

Apache Software Foundation: Jakarta JetSpeed 1.3

JetSpeed home page: http://jakarta.apache.org/jetspeed/site/index.html

JetSpeed Portlet API: http://cvs.apache.org/viewcvs/jakarta-jetspeed/proposals/portletAPI/

BEA: Web Logic Portal 4.0 http://www.bea.com/products/weblogic/portal/index.shtml

IBM: WebSphere Portal 2.1 http://www-4.ibm.com/software/webservers/portal/

iPlanet: iPlanet Portal Server 3.0
http://www.iplanet.com/products/iplanet_portal/home_portal.html

Oracle: Oracle 9i Portal http://www.oracle.com/ip/deploy/ias/portal/index.html

3.2 Explanation of how these items might be used as a starting point for the work.

They will be useful for gathering features and evaluating the effectiveness and shortcoming of each implementation.

Search

Introduction | JSRs | What's New | Participation
JCP Procedures | Press & Success | Community Resources