

Printed: Feb 20, 2002

From: <http://jcp.org/jsr/detail/127.prt>

[Standard Format](#)

JSRs: Java Specification Requests

Detail

JSR 127

JavaServer Faces

This specification defines an architecture and APIs which simplify the creation and maintenance of Java Server application GUIs.

Specification Lead

[Amy Fowler](#)

Sun Microsystems, Inc.

[Hans Muller](#)

Sun Microsystems, Inc.

Expert Group

Aligo, Inc.

Art Technology Group Inc.(ATG)

BEA Systems

Borland Software Corporation

Developmentor

EDS

Hewlett-Packard

ILOG

Lazarus, Eric

Macromedia, Inc.

Murrish, Randy

Netdecisions Holdings United

SAS Institute

Apache Software Foundation

Bayern, Shawn

Bogaert, Mathias

Carapetyan, Pete

Documentum, Inc.

Fujitsu Limited

IBM

IONA Technologies PLC

Lutris Tecnologies

Mettu, Kumar

Nash, Michael

Oracle

Siemens AG

Silverstream Software
Sun Microsystems, Inc.
Zukowski, John

Strachan, James
WebGain

[Private Expert Group homepage](#) 

Status

Expert Group Formation	30 May, 2001
JSR Review Ballot	29 May, 2001

Java Community Process version in use: [2.1](#)

Please direct your comments on this JSR to jsr-127-comments@jcp.org

Updates to the Original Java Specification Request (JSR)

Current JSR Status

The JSR127 expert group is actively developing the Community draft of the JavaServer Faces specification. We are targeting the Community Draft for Q1CY-02, the Public draft for Q2CY-02, and the Final draft for Q3CY-02. Note: these targets are subject to change by the expert group.

Along with the Community and Public drafts of the specification, we plan to release early access versions of the reference implementation in order to get developer feedback before the specification goes to Final draft. As soon as we have a target date available for an early access release we will make that information available on this page.

We'll look forward to getting your comments on the technology when the spec drafts and reference implementation snapshots are released for your review.

Thank you,
The JSR127 Expert Group

Original Java Specification Request (JSR)

[Identification](#) | [Request](#) | [Contributions](#)

Section 1. Identification

Submitting Member: Sun Microsystems

Name of Contact Person: Amy Fowler

E-Mail Address: amy.fowler@eng.sun.com

Telephone Number: +1 408 343 1418

Fax Number: +1 408 343 1797

Specification Lead: Amy Fowler, Hans Muller, Java Software, Sun Microsystems

E-Mail Address: amy.fowler@eng.sun.com, hans.muller@eng.sun.com

Telephone Number: +1 408 343 1418

Fax Number: +1 408 343 1797

Initial Expert Group Membership:

ATG
Borland
Documentum
Macromedia
SAS
Sitraka
Sun Microsystems
WebGain

Section 2: Request

2.1 Please describe the proposed Specification:

The Servlet/JSP environment provides a powerful model for creating web applications, however it defines no APIs specifically for creating the client GUI. To build a JSP page that contains one or more HTML forms, a developer must manage the form's GUI state and build a mechanism to dispatch from incoming HTTP requests to component specific event handling methods. When constructing a web site that provides a complex GUI, developers often create special purpose infrastructure that simplifies re-using form parts and facilitates the process of applying sweeping style and behavioral changes to all of the GUI elements on the site. Over time, the specialized infrastructure created to support all of these tasks tends to take on a life of it's own. Unfortunately the idiosyncratic qualities of a particular site's GUI software make it difficult to take advantage of generic level programming tools and environments. Ideally an off the shelf tool would simplify high level tasks like arranging and reusing GUI components and interconnecting component state and input handling with the objects that define the application's behavior.

This is a proposal to define a standard set of JSP tags and Java classes that simplify building Java Server application GUIs. Beyond addressing a common problem, creating a standard way to define complex HTML forms and other common GUI elements will enable tools vendors and third party component vendors to focus their energy on a single component framework for JSP/servlets. It will also help bridge the gap between conventional GUI toolkit developers and web based GUI developers by providing familiar APIs for GUI components, component state, and for rendering and input processing. Comprehensive support for internationalization and basic input validation will ensure that developers include these features in the first release of their applications.

The following 8 design goals represent the focus of this proposal:

1. Create a standard GUI component framework which can be leveraged by development tools to make it easier for tool users to both create high quality GUIs and manage the GUI's connections to application behavior.
2. Define a set of simple lightweight Java base classes for GUI components, component state, and input events. These classes will address GUI lifecycle issues, notably managing a component's persistent state for the lifetime of its page.
3. Provide a set of common GUI components, including the standard HTML form input elements. These components will be derived from the simple set of base classes

(outlined in #1) that can be used to define new components.

4. Provide a JavaBeans model for dispatching events from client-side GUI controls to server-side application behavior.
5. Define APIs for input validation, including support for client-side validation.
6. Specify a model for internationalization and localization of the GUI.
7. Automatic generation of appropriate output for the target client, taking into account all available client configuration data, such as browser version, etc.
8. Automatic Generation of output containing required hooks for supporting accessibility, as defined by WAI.

It is understood that many of the problem spaces targeted by this JSR are already being addressed in other projects. One prime example of such a project is the Jakarta open source project, Struts. However, this JSR is aimed at creating a standard for Java server application GUIs, which will serve to unify what's become a fragmented area and strengthen support in tools and other third party products. The expert group will work to determine where there are synergies and overlap with existing efforts, and ensure this specification draws appropriately from such projects.

This JSR is viewed as synergistic with the JSP Standard Tag Library project (JSPTL, JSR-052), as JSPTL does not include GUI-related APIs.

Additionally, work continues in the W3C to evolve relevant web technologies (XHTML, XForms) and the expert group will track these efforts to ensure the proposed specification includes enough flexibility to evolve with these technologies as needed.

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

J2EE

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

This specification will establish a standard API for creating Java Web application GUIs, which will eliminate the burden of creating and maintaining GUI infrastructure for developers.

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

The Servlet/JSP APIs define no GUI structure beyond emitting the appropriate HTML for the client. There is no model for representing GUI elements as manipulable, stateful objects on the server, and no standard mechanism for dispatching client events back to specific methods in the server application. There are no standard APIs for supporting internationalization, localization, or accessibility.

The existing Java specifications for creating GUIs, AWT and Swing, were not designed for modeling GUI elements within an HTML page across an HTTP connection, which is a scenario requiring significantly different rendering and event models. However, since a design goal of this JSR is to provide a GUI paradigm which is familiar to developers, applicable designs and APIs, such as the JavaBeans event model, will be drawn from these and other existing GUI toolkits.

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

See Specification description in Section 2.1.

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

`javax.servlet.ui`

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 to support internationalization and localization are a key 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 such changes are anticipated. The existing GUI specifications, AWT and Swing, are targeted for traditional object-oriented GUI development and will continue to be supported.

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

The final schedule will be determined by the expert group, however the target for the Community Draft specification is Fall 2001.

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

Most of the working group communication will happen via email discussion. There will be a face-to-face meeting to initiate the working group activity.

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.

Specifications for the Java Server development environment:

JSP 1.1 Specification (<http://java.sun.com/products/jsp>)

Servlet 2.2 Specification (<http://java.sun.com/products/servlet>)

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

This JSR specification will include a new Java class library which hooks in appropriately with the Servlet 2.2 API and an associated JSP Tag library based on the JSP 1.1 specification. Note that the revisions to these APIs, Servlet 2.3 and JSP 1.2, are currently in progress, however it is not anticipated that this JSR will have any dependencies on functionality introduced in these revisions.

[Introduction](#) | [JSRs](#) | [What's New](#) | [Participation](#)
[JCP Procedures](#) | [Press & Success](#) | [Community Resources](#)



Site sponsored and powered by [Sun Microsystems](#)

Copyright © 1995-2002. All Rights Reserved.

[Terms of Use](#). [Privacy Policy](#).