

JavaServer Pages[™] Technology

*Powering the Web Experience with Dynamic
Content*



Sun Microsystems, Inc.
901 San Antonio Road
Palo Alto, CA 94303
1 (800) 786.7638
1.512.434.1511

Copyright 1999 Sun Microsystems, Inc., 901 San Antonio Road, Palo Alto, California 94303 U.S.A. All rights reserved.

This product or document is protected by copyright and distributed under licenses restricting its use, copying, distribution, and decompilation. No part of this product or document may be reproduced in any form by any means without prior written authorization of Sun and its licensors, if any. Third-party software, including font technology, is copyrighted and licensed from Sun suppliers.

Parts of the product may be derived from Berkeley BSD systems, licensed from the University of California. UNIX is a registered trademark in the U.S. and other countries, exclusively licensed through X/Open Company, Ltd.

Sun, Sun Microsystems, the Sun logo, Java, JavaBeans, Enterprise JavaBeans, JavaMail, JavaServer Pages, JDBC, and iPlanet are trademarks, registered trademarks, or service marks of Sun Microsystems, Inc. in the U.S. and other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc. Netscape is a trademark or registered trademark of Netscape Communications Corporation in the United States and other countries.

The OPEN LOOK and Sun™ Graphical User Interface was developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

RESTRICTED RIGHTS: Use, duplication, or disclosure by the U.S. Government is subject to restrictions of FAR 52.227-14(g)(2)(6/87) and FAR 52.227-19(6/87), or DFAR 252.227-7015(b)(6/95) and DFAR 227.7202-3(a).

DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

Copyright 1999 Sun Microsystems, Inc., 901 San Antonio Road, Palo Alto, Californie 94303 Etats-Unis. Tous droits réservés.

Ce produit ou document est protégé par un copyright et distribué avec des licences qui en restreignent l'utilisation, la copie, la distribution, et la décompilation. Aucune partie de ce produit ou document ne peut être reproduite sous aucune forme, par quelque moyen que ce soit, sans l'autorisation préalable et écrite de Sun et de ses bailleurs de licence, s'il y en a. Le logiciel détenu par des tiers, et qui comprend la technologie relative aux polices de caractères, est protégé par un copyright et licencié par des fournisseurs de Sun.

Des parties de ce produit pourront être dérivées des systèmes Berkeley BSD licenciés par l'Université de Californie. UNIX est une marque déposée aux Etats-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company, Ltd.

Sun, Sun Microsystems, le logo Sun, Java, JavaBeans, Enterprise JavaBeans, JavaMail, JavaServer Pages, JDBC, and iPlanet sont des marques de fabrique ou des marques déposées, ou marques de service, de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays. Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc. Netscape est une marque de Netscape Communications Corporation aux Etats-Unis et dans d'autres pays.

L'interface d'utilisation graphique OPEN LOOK et Sun™ a été développée par Sun Microsystems, Inc. pour ses utilisateurs et licenciés. Sun reconnaît les efforts de pionniers de Xerox pour la recherche et le développement du concept des interfaces d'utilisation visuelle ou graphique pour l'industrie de l'informatique. Sun détient une licence non exclusive de Xerox sur l'interface d'utilisation graphique Xerox, cette licence couvrant également les licenciés de Sun qui mettent en place l'interface d'utilisation graphique OPEN LOOK et qui en outre se conforment aux licences écrites de Sun.

CETTE PUBLICATION EST FOURNIE "EN L'ETAT" ET AUCUNE GARANTIE, EXPRESSE OU IMPLICITE, N'EST ACCORDEE, Y COMPRIS DES GARANTIES CONCERNANT LA VALEUR MARCHANDE, L'APTITUDE DE LA PUBLICATION A REpondre A UNE UTILISATION PARTICULIERE, OU LE FAIT QU'ELLE NE SOIT PAS CONTREFAISANTE DE PRODUIT DE TIERS. CE DENI DE GARANTIE NE S'APPLIQUERAIT PAS, DANS LA MESURE OU IL SERAIT TENU JURIDIQUEMENT NUL ET NON AVENU.



Please
Recycle



Adobe PostScript

Introduction

Java™ Servlet technology has done much towards making Web application development an easier task. With a platform- and server-independent approach based on the Java™ language, servlets provide developers with an easier way to build high-performance, dynamic Web pages compared to traditional CGI programs.

With JavaServer Pages™ (JSP) technology, Sun makes creating dynamic HTML and XML Web pages even simpler. JSP technology is a major extension of Java Servlet technology that makes building and maintaining dynamic pages much easier, and supports the distributed development model common in larger enterprises. JSP technology opens up dynamic pages to a wide range of page authors, while offering all of the benefits of the Java platform.

This paper describes how JSP technology extends the basic servlet framework to support easier and faster page authoring and maintenance. For those familiar with or already using servlets, incorporating the additional functionality of JSP technology is fast and easy. To find out more about how JSP pages fit with what you already do, please see the *JavaServer Pages Technology for Servlet Developers* paper.

Java Servlets: Building the Foundation for Dynamic Web Applications

As the server-side counterpart of an applet, Java servlets have made many Web applications possible. With Java servlets, Web developers get a simple, consistent mechanism for extending the functionality of a Web server and accessing existing business systems.

Today, servlets are a popular choice for building interactive Web applications. Third-party servlet containers are available for the Apache Web server, iPlanet™ Web Server (formerly Netscape™ Enterprise Server), Microsoft IIS, and others. Servlet containers can also be integrated with Web-enabled application servers such as the BEA WebLogic application server, IBM WebSphere, Netscape Application Server, and more.

A major advantage of servlets is that they provide a component-based, platform-independent method for building Web-based applications — without the performance limitations of CGI programs. And unlike proprietary server extension mechanisms (such as the Netscape Server API or Apache modules), servlets are both server and platform independent. This enables companies to select a best-of-breed strategy for their servers, platforms, and tools.

Written in the Java language, servlets have access to the entire family of Java APIs, including the JDBC™ API, to access enterprise databases. Servlets can also access a library of HTTP-specific calls, and offer all the benefits of the mature Java language, including portability, performance, reusability, and crash protection. But for all of their benefits, servlets are still primarily a developer's tool. With JSP technology, the capabilities of servlets have been extended to a broader audience of page authors.

JavaServer Pages: Authoring Pages Instead of Writing Programs

A JSP page is actually compiled into a servlet when it is called, so JSP technology is truly an extension of servlet technology. However, JSP pages make servlets much easier to write and maintain by changing the approach to creating servlets.

A page created with JSP technology can be thought of as an “inside-out” servlet. In a servlet, fixed page elements (such as text) are embedded in the application logic (for example, in an `out.println` statement). Between the statements and the other programming overhead (such as preambles and boilerplates), it can be difficult to even find the fixed elements, much less change them.

In contrast, a JSP page embeds the application logic within standard HTML (or XML) templates. Page authors can create or edit page design elements or text without affecting application logic at all. Because the JSP format automatically takes care of the programming overhead, and the JSP container automatically compiles the page when it is called, the process of creating and revising pages is greatly simplified versus that of writing and compiling servlet programs.

The application logic, which is embedded in the JSP page, may be in one of the following forms:

- **Java scriptlets:** Script fragments that perform simple processing
- **JSP tags that call external functions,** for example, a `USEBEAN` tag gives the page author easy access to a JavaBeans™ component that may perform some basic functionality
- **Customized JSP tags:** Developers can create customized tag libraries that page authors may call with a simple, XML-like syntax

Easier and Faster Page Authoring

JSP technology simplifies and speeds the development process for developers and page authors alike by separating code development and page layout activities.

- Instead of writing a program, page authors simply write a page using traditional HTML, then add the XML-like tags and, if necessary, scriptlets to tie everything together. Most of the programming overhead is automatically taken care of by the JSP container.
- The pages automatically compile into servlets when called, so the page author does not need to worry about compiling changes. JSP pages can also be precompiled into servlets.
- Developers can write beans, servlets, or customized tag libraries, and then distribute them to a wide range of page authors. This makes everyone more efficient – developers can concentrate on developing and everyone else can make use of their efforts, quickly and easily.
- By supporting component-based development and customized tag libraries, JSP technology not only simplifies page authoring but also provides a strong foundation for a wide range of page authoring tools, which are now becoming more and more available.

Simplified Application Maintenance Through Separation of Content and Logic

JSP pages are not only fast to deploy, they are easy to maintain over time. This is due to the separation of application logic (typically residing within customized tag libraries or beans) from page design/content.

For example:

- A page author can change fixed text on the page without affecting the application logic at all. The JSP container will recompile the page automatically the first time it is called.
- Likewise, application developers can create, distribute, and edit application logic through customized JSP tag libraries. They can revise components without changing the individual pages that access the components.

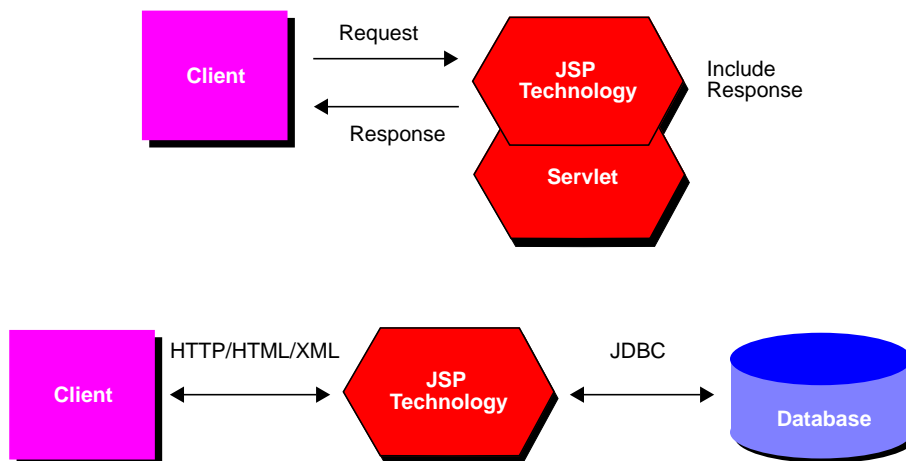
Using JSP technology, individuals responsible for pages can make content changes without requiring a developer's time. This leads to faster application changes and overall easier application maintenance.

JSP Pages Fit Within Many Architectures

There are many ways to use JSP pages within a broader Web-based application. As part of the Java family, JSP technology has access to all of the Java APIs and components. For example, a JSP page may include a request of a servlet to process information. The JSP page could determine details of the request, and then forward the request appropriately based on the request properties.

A JSP page may access other data sources directly, using the appropriate Java invocation method. JSP pages and servlets can access relational databases through the JDBC API. In the future, custom tags will handle the database management functions.

The architecture diagrammed below is a simple method for accessing data sources, and suits relatively low-volume applications.



JSP pages may call other Java components, such as Enterprise JavaBeans™ (EJB) components. For example, an EJB component and container can manage access to external resources, providing better performance for high-volume applications. By implementing custom JSP tags to support calls to these resources, the page author can be shielded from the details of the enterprise environment.



A Java Solution for the Enterprise

JSP and Java Servlet technologies offer all the advantages of the Java language, a mature, simplified solution suitable for both large and small applications. It offers:

- Portability across platforms and servers
- Robust exception management and memory management
- Access to the wide range of Java APIs (JDBC, JavaMail™, etc.)

JSP and Java Servlet technologies are part of the Java 2 Platform, Enterprise Edition (J2EE), and provide a highly flexible and scalable solution for building multitier applications in the enterprise environment because they:

- Can access Enterprise JavaBeans and J2EE components and containers that package a wide range of application functionality, including security and transaction management
- Provide multithreaded processing and multiple concurrent requests, supporting applications with large numbers of concurrent users

In conclusion, using JSP and Java Servlet technologies helps leverage all the benefits of Java technology for rapid application deployment and easier application maintenance. Servlets and JSP pages can support large-scale interactive Web applications in the enterprise environment with a distributed development model. Web developers can create cross-platform application components as JavaBeans, Enterprise JavaBeans, or customized tag libraries. And page authors can use a simplified interface to integrate this logic into their page designs.



Sun Microsystems, Inc.
901 San Antonio Road
Palo Alto, CA 94303

1 (800) 786.7638
1.512.434.1511

<http://java.sun.com>

November 1999