

Introduction



Suppose your company wants to sell products online. You have a database that gives the price and inventory status of each item. However, your database doesn't speak HTTP, the protocol that Web browsers use. Nor does it output HTML, the format Web browsers need. What can you do? Once users know what they want to buy, how do you gather that information? You want to customize your site for visitors' preferences and interests, but how? You want to keep track of user's purchases as they shop at your site, but what techniques are required to implement this behavior? When your Web site becomes popular, you might want to compress pages to reduce bandwidth. How can you do this without causing your site to fail for those visitors whose browsers don't support compression? In all these cases, you need a program to act as the intermediary between the browser and some server-side resource. This book is about using the Java platform for this type of program.

"Wait a second," you say. "Didn't you already write a book about that?" Well, yes. In May of 2000, Sun Microsystems Press and Prentice Hall released Marty Hall's second book, *Core Servlets and JavaServer Pages*. It was successful beyond everyone's wildest expectations, selling approximately 100,000 copies, getting translated into Bulgarian, Chinese simplified script, Chinese traditional script, Czech, French, German, Hebrew, Japanese, Korean, Polish, Russian, and Spanish, and being chosen by Amazon.com as one of the top five computer programming books of 2001. What fun!

Since then, use of servlets and JSP has continued to grow at a phenomenal rate. The Java 2 Platform has become the technology of choice for developing e-commerce applications, dynamic Web sites, and Web-enabled applications and service. Servlets and JSP continue to be the foundation of this platform—they provide the link between Web clients and server-side applications. Virtually all major

Web servers for Windows, UNIX (including Linux), Mac OS, VMS, and mainframe operating systems now support servlet and JSP technology either natively or by means of a plug-in. With only a small amount of configuration, you can run servlets and JSP in Microsoft IIS, the Apache Web Server, IBM WebSphere, BEA WebLogic, Oracle Application Server 10g, and dozens of other servers. Performance of both commercial and open-source servlet and JSP engines has improved significantly.

To no one's surprise, this field continues to grow at a rapid rate. As a result, we could no longer cover the technology in a single book. *Core Servlets and JavaServer Pages, Volume 1: Core Technologies*, covers the servlet and JSP capabilities that you are likely to use in almost every real-life project. This book, *Volume 2: Advanced Technologies*, covers features that you may use less frequently but are extremely valuable in robust applications. For example,

- **Deployment descriptor file.** Through the proper use of the deployment descriptor file, `web.xml`, you can control many aspects of the Web application behavior, from preloading servlets, to restricting resource access, to controlling session time-outs.
- **Web application security.** In any Web application today, security is a must! The servlet and JSP security model allows you to easily create login pages and control access to resources.
- **Custom tag libraries.** Custom tags significantly improve the design of JSPs. Custom tags allow you to easily develop your own library of reusable tags specific to your business applications. In addition to creating your own tags, we cover the Standard Tag Library (JSTL).
- **Event handling.** With the events framework, you can control initialization and shutdown of the Web application, recognize destruction of HTTP sessions, and set application-wide values.
- **Servlet and JSP filters.** With filters, you can apply many pre- and post-processing actions. For instance, logging incoming requests, blocking access, and modifying the servlet or JSP response.
- **Apache Struts.** This framework greatly enhances the standard model-view-controller (MVC) architecture available with servlets and JSPs. More importantly, Apache Struts still remains one of the most common frameworks used in industry.

Who Should Read This Book

The main audience is developers who are familiar with basic servlet and JSP technologies, but want to make use of advanced capabilities. As we cover many topics in this book—the deployment descriptor file, security, listeners, custom tags, JSTL, Struts,

Ant—you may want to first start with the technologies of most interest, and then later read the remaining material. Most commercial servlet and JSP Web applications take advantage of the technologies presented throughout, thus, at some point you may want to read the complete book.

If you are new to servlets and JSPs, you will want to read *Core Servlets and Java-Server Pages, Volume 1: Core Technologies*. In addition to teaching you how to install and configure a servlet container, Volume 1 provides excellent coverage of the servlet and JSP specifications. Volume 1 provides the foundation material to this book.

Both books assume that you are familiar with basic Java programming. You don't have to be an expert Java developer, but if you know nothing about the Java programming language, this is not the place to start. After all, servlet and JSP technology is an application of the Java programming language. If you don't know the language, you can't apply it. So, if you know nothing about basic Java development, start with a good introductory book like *Thinking in Java*, *Core Java*, or *Core Web Programming*, all from Prentice Hall.

Conventions

Throughout the book, concrete programming constructs or program output are presented in a monospaced font. For example, when abstractly discussing server-side programs that use HTTP, we might refer to “HTTP servlets” or just “servlets,” but when we say `HttpServlet` we are talking about a specific Java class.

User input is indicated in boldface, and command-line prompts are either generic (`Prompt>`) or indicate the operating system to which they apply (`DOS>`). For instance, the following indicates that “Some Output” is the result when “`java SomeProgram`” is executed on any platform.

```
Prompt> java SomeProgram  
Some Output
```

URLs, file names, and directory names are presented in a sans serif font. So, for example, we would say “the `StringTokenizer` class” (monospaced because we're talking about the class name) and “Listing such and such shows `SomeFile.java`” (sans-serif because we're talking about the file name). Paths use forward slashes as in URLs unless they are specific to the Windows operating system. So, for instance, we would use a forward slash when saying “look in `install_dir/bin`” (OS neutral), but use backslashes when saying “see `C:\Windows\Temp`” (Windows specific).

Important standard techniques are indicated by specially marked entries, as in the following example.



Core Approach

Pay particular attention to items in Core Approach sections. They indicate techniques that should always or almost always be used.

Core Notes and Core Warnings are called out in a similar manner.

About the Web Site

The book has a companion Web site at <http://volume2.coreservlets.com/>. This free site includes:

- Documented source code for all examples shown in the book, which can be downloaded for unrestricted use.
- Links to all URLs mentioned in the text of the book.
- Up-to-date download sites for servlet and JSP software.
- Information on book discounts.
- Book additions, updates, and news.