Servlets

Servlets are the Java programs that run on the Java-enabled web server or application server. They are used to handle the request obtained from the webserver, process the request, produce the response, and then send a response back to the webserver.

Chapter 3 Migrating Java EE Applications

You use the Migration Tool or the asmigrate command to migrate applications from competitive application servers. You also use this tool to migrate the applications that do not deploy successfully after upgrading from an older version of Sun Java System Application Server. This tool works on the input archive or source code to translate the runtime deployment descriptors from the source application server format to generate runtime deployment descriptors that are compliant with the latest version. It also parses the JSP and Java source code files (in case of source code input) and provides runtime support for certain custom JSP tags and proprietary APIs.

This chapter addresses the following topics:

- Understanding Migration
- Deploying the Migrated Application

Understanding Migration

This section describes the need to migrate Java EE applications and the particular files that must be migrated. Following successful migration, a Java EE application is redeployed to the Application Server.

The following topics are addressed:

- Java EE Components and Standards
- Java EE Application Components
- Why is Migration Necessary?
- What Needs to be Migrated
- Migration Tool and Other Resources

Java EE Components and Standards

Sun Java System Application Server 9.1 (hereafter called Application Server) is a Java EE-compliant server based on the component standards developed by the Java community. By contrast, the Sun Java System Application Server 7 (Application Server 8) is a J2EE v1.3-compliant server and Sun ONE Application Server 6.x (Application Server 6.x) is a J2EE v1.2-compliant server. Between the four versions, there are considerable differences with the application component APIs.

The following table characterizes the differences between the component APIs used with the J2EE v1.4-compliant Sun Java System Application Server 9.1, the J2EE v1.3-compliant Sun ONE Application Server 7, and the J2EE v1.2-compliant Sun ONE Application Server 6.x.

Table 3–1 Application Server Version Comparison of APIs for Java EE Components

Component API	**Sun ONE Application Server 6.x**	**Sun Java System Application Server 7**	**Sun Java System Application Server 8.2**	**Sun Java System Application Server 9.1**
JDK	1.2.2	1.4	1.4	
Servlet	2.2	2.3	2.4	
JSP	1.1	1.2	2.0	
JDBC	2.0	2.0	2.1, 3.0	
EJB	1.1	2.0	2.0	
JNDI	1.2	1.2	1.2.1	
JMS	1.0	1.1	1.1	
JTA	1.0	1.01	1.01	

Java EE Application Components

Java EE simplifies development of enterprise applications by basing them on standardized, modular components, providing a complete set of services to those components, and handling many details of application behavior automatically, without complex programming. J2EE v1.4 architecture includes several component APIs. Prominent Java EE components include:

- Client Application
- Web Application
- Enterprise Java Beans (EJB)

- Connector
- Enterprise Application Archive (EAR)

Java EE components are packaged separately and bundled into a Java EE application for deployment. Each component, its related files such as GIF and HTML files or server-side utility classes, and a deployment descriptor are assembled into a module and added to the Java EE application. A Java EE application is composed of one or more enterprise bean(s), Web, or application client component modules. The final enterprise solution can use one Java EE application or be made up of two or more Java EE applications, depending on design requirements.

A Java EE application and each of its modules has its own deployment descriptor. A deployment descriptor is an XML document with a .xml extension that describes a component's deployment settings.

A Java EE application with all of its modules is delivered in an Enterprise Archive (EAR) file. An EAR file is a standard Java Archive (JAR) file with a .ear extension. The EAR file contains EJB JAR files, application client JAR files and/or Web Archive (WAR) files.

For more information on Java EE, see Java EE website:

Why is Migration Necessary?

Although Java EE specifications broadly cover requirements for applications, they are nonetheless evolving standards. They either do not cover some aspects of applications or leave implementation details to the application providers.

This leads to different implementations of the application servers, also well as difference in the deployment of Java EE components on application servers. The array of available configuration and deployment tools for use with any particular application server product also contributes to the product implementation differences.

The evolutionary nature of the specifications itself presents challenges to application providers. Each of the component APIs are also evolving. This leads to a varying degree of conformance by products. In particular, an emerging product, such as the Application Server, has to contend with differences in Java EE application components, modules, and files deployed on other established

application server platforms. Such differences require mappings between earlier implementation details of the Java EE standard, such as file naming conventions, and messaging syntax.

Moreover, product providers usually bundle additional features and services with their products. These features are available as custom JSP tags or proprietary Java API libraries. Unfortunately, using these proprietary features renders these applications non-portable.

What Needs to be Migrated

The Java EE application consists of the following file categories that need to be migrated:

- Deployment descriptors (XML files)
- JSP source files that contain Proprietary APIs
- Java source files that contain Proprietary APIs

Deployment descriptors (XML files)

Deployment is accomplished by specifying deployment descriptors for standalone enterprise beans (EJB, JAR files), front-end Web components (WAR files) and enterprise applications (EAR files). Deployment descriptors are used to resolve all external dependencies of the Java EE components or applications. The Java EE specification for deployment descriptors is common across all application server products. However, the specification leaves several deployment aspects of components pertaining to an application dependent on product implementation.

JSP source files

Java EE specifies how to extend JSP by adding extra custom tags. Product vendors include some custom JSP extensions in their products, simplifying some tasks for developers. However, usage of these proprietary custom tags results in non-portability of JSP files. Additionally, JSP can invoke methods defined in other Java source files as well. The JSPs containing proprietary APIs need to be rewritten before they can be migrated.

Java source files

The Java source files can be EJBs, servlets, or other helper classes. The EJBs and servlets can invoke standard Java EE services directly. They can also invoke methods defined in helper classes.

Java source files are used to encode the business layer of applications, such as EJBs. Vendors bundle several services and proprietary Java API with their products. The use of proprietary Java APIs is a major source of non-portability in applications. Since Java EE is an evolving standard, different products can support different versions of Java EE component APIs.

Migration Tool and Other Resources

The Migration Tool for Sun Java System Application Server 9.1 (hereafter called Migration Tool) migrates Java EE applications from other server platforms to Sun Java System Application Server 9.1.

The following source platforms are supported for Sun Java System Application Server 9.1:

- Sun ONE Application Server 6.*x*
- Sun Java System Application Server 7
- Sun Java System Application Server 8.0/8.1
- Java EE Reference Implementation Application Server (RI) 1.3, 1.4 Beta1
- WebLogic Application Server (WLS) 5.1, 6.0, 6.1, 8.1
- WebSphere Application Server (WAS) 4.0, 5.x
- Sun ONE Web Server 6.0
- JBoss Application Server 3.0
- TomCat Web Server 4.1

Migration Tool automates the migration of Java EE applications to Sun Java System Application Server 9.1, without much modification to the source code.

The key features of the tool are:

- Migration of application server-specific deployment descriptors
- Runtime support for selected custom Java Server Pages (JSP) tags and proprietary APIs
- Conversion of selected configuration parameters with equivalent functionality in Application Server
- Automatic generation of Ant based scripts for building and deploying the migrated application to the target server, Application Server
- Generation of comprehensive migration reports after achieving migration

Download the Migration Tool from the following location:

http://java.sun.com/j2ee/tools/migration/index.html.

The Java Application Verification Kit (AVK) for the Enterprise helps build and test applications to ensure that they are using the J2EE APIs correctly and to migrate to other J2EE compatible application servers using specific guidelines and rules.

Download the Java Application Verification Kit (AVK) from the following location:

http://java.sun.com/j2ee/verified/.

Deploying the Migrated Application

To be able to deploy your migrated applications on Application Server 9.1, it is important to understand classloaders in Application Server 9.1 and changes to the architecture.

Application Server 8.x does not support overriding of libraries such as the default parser. Application Server 9.1 provides the **`-libraries`** option for overriding the default XML parser and using a different JAXP compatible implementation of the parser.

You can use the **`-libraries`** option to control the scope of your libraries. See the Chapter 2, *Class Loaders*, in *Sun Java System Application Server 9.1 Developer's Guide* for a detailed description of the classloaderhandling mechanism in Application Server 9.1. For more on classloader delegation in Application Server 9.1, see *Class Loader Delegation* in *Sun Java System Application Server 9.1 Developer's Guide*.