Sun Java™ System Application Server Platform Edition 8 Release Notes

March 2004

Part Number 817-6082

The Sun Java™ System Application Server Platform Edition 8 product greatly simplifies the task of creating and administering web services applications. The Application Server provides a development path for web services that simplifies the development process while providing uniquely flexible growth opportunities.

These release notes contain important information available at the time of the Sun Java™ System Application Server Platform Edition 8 product release. Product requirements, what's new, platform summary, known problems, and other late-breaking issues are addressed here. Read this document before you begin using the Application Server product.

This document contains the following sections:

- What's New in the Application Server
- Key Features
- System Requirements
- Documentation
- Known Problems and Limitations
- How to Report Problems
- For More Information
- Revision History

What's New in the Application Server

The Sun Java System Application Server Platform Edition 8 implements the new features specified by the J2EE 1.4 platform. The following sections describe the new features and technologies:

- J2EE 1.4 Support
- Reduced Download Size and Memory Footprint
- Enhanced Developer Experience, Usability and Productivity
- Automatic Deployment
- Log Viewer
- JavaServer Faces Technology
- J2EE SDK Bundle

J2EE 1.4 Support

The Sun Java System Application Server Platform Edition 8 supports the J2EE 1.4 platform; the most complete platform for Web services in the market today. The J2EE 1.4 platform supports WS-I Basic Profile 1.0 and enables Web services to interoperate on the basis of Web Service Description Language (WSDL) and Simple Object Access Protocol (SOAP). Its many features include:

- Exposure of J2EE applications as Web services that are based on SOAP and Hypertext Transfer Protocol (HTTP)
- Integration with Web services and implementation of Web-service endpoints with Java servlets and Enterprise Java Beans (EJB) technology
- New APIs for Web services, deployment, and management of the J2EE platform, as well as authorization
- Updated APIs, including EJB, Java servlet, JavaServer Pages (JSP), and J2EE connector APIs

The following table describes the enhanced APIs available on the J2EE 1.4 platform.

Table 1   Major API changes on the J2EE 1.4 Platform

API	Description
Components	
Application and Application Client	Implementation of standard deployment descriptors by means of XML schemas
Enterprise JavaBeans (EJB) 2.1	Time service and EJB Web-service endpoint
Java Servlet 2.4	Web-service endpoint filter
JavaServer Pages (JSP) 2.0 architecture	Expression language and tag library
J2EE Connector Architecture 1.5	Inbound resource adaptor and Java Message Service (JMS) pluggability

Table 1 Major API changes on the J2EE 1.4 Platform

API	Description
Web Services	
Web services for J2EE 1.1	Portable Web-service descriptors
Java API for XML-based Remote Procedure Calls (JAX-RPC) 1.1	Mapping for WSDL and Java technology and support for development of Web-service clients and endpoints
WS-I Basic Profile 1.0	The enabling element for interoperability between WSDL and SOAP
SOAP with attachment API for Java (SAAJ) 1.2	An API for SOAP-based messaging; fosters the creation of SOAP messages with attachments
Java APIs for XML Registries (JAXR) 1.0	A uniform and standard API for accessing XML registries, such as those for Universal Description Discovery and Integration (UDDI and ebXML)
Other	
J2EE application Deployment 1.1	Standard APIs that enable deployments of J2EE components and applications
J2EE Management 1.0	Definitions for the information model for managing the J2EE platform
Java Management Extensions (JMX) 1.2	Management of standards
Java Authorization Contract for Containers (JACC) 1.0	Definitions of security contracts between a J2EE application server and the authorization policy provider
Java API for XML Processing (JAXP) 1.2	An API with which applications can parse and transform XML documents; also adds support for processing of XML schemas
JMS 1.1	A messaging standard that enables J2EE application components to create, send, receive, and read messages; also adds support for uniform APIs for queues and topics
JavaMail 1.3	A set of abstract classes that model a mail system; also includes minor updates to the APIs

Reduced Download Size and Memory Footprint

The download size for the Sun Java System Application Server Platform Edition 8, with only a 36-Mbyte footprint, represents a major reduction when compared with that of Application Server 7. The requirements for startup memory stands at only 163 Mbytes—a much faster startup cycle and a significant improvement over Application Server 7.

Another contributor to an efficient cycle is that only two processes are loaded at startup: one for the Sun Java System Application Server Platform Edition 8, and another for Sun Java System Message Queue.

Enhanced Developer Experience, Usability and Productivity

The Sun Java System Application Server 8 Platform Edition provides the following developer tools:

- **Administration Console** — for administering and configuring the Application Server Platform Edition 8 from the user interface. Specifically, you can setup the following from the Administration Console:

  - Java Database Connectivity (JDBC), JMS, and connector resources

  - User and security realms

  - Configuration and tuning parameters for containers and services

  - Logging and monitoring levels of subsystems

- **Deploy Tool** — for assembling J2EE components into applications and deploying those applications to the Sun Java System Application Server Platform Edition 8.

- **J2EE Application Verifier** — for performing predeployment checks on applications to ensure that their syntaxes and packaging are correct.

- **Upgrade Tool** — for migrating configuration files from Application Server 7 Platform Edition.

- **Migration Tool** — for migrating applications from BEA, IBM, and JBoss. You can download Migration Tool separately at http://iforce.sun.com/partners/migration/

- **Apache Ant 1.5** — for building J2EE applications. Application Server Platform Edition 8 also includes a number of custom Ant tasks that simplify deployment and administration from Ant build files.

Automatic Deployment

You can automatically deploy an application in a running instance of the Application Server by copying the EAR, EJB JAR, or WAR file to the domain's `autodeploy` directory.

Log Viewer

The Log Viewer developer tool displays messages and processes. The logging capability, based on the J2SE logging API, boasts a new, uniform format and viewer. You can change the log level on a subsystem basis without restarting the server, and you can also add custom log handlers and filters.

JavaServer Faces Technology

The Sun Java System Application Server 8 Platform Edition is the first middleware product that bundles and supports JavaServer Faces technology. The JavaServer Faces technology consists of a set of server-side APIs that represent user-interface components that manage their state, event, handling, and input validation. The APIs also define page navigation and support internationalization and accessibility. You can add custom UI components with a JSP custom tag library.

While developing with JavaServer Faces technology, each member of a development team can focus on a single piece of the process. A simple programming model then links the pieces, resulting in a much more efficient and simpler development cycle.

J2EE SDK Bundle

For this release, you can get a stand-alone Sun Java System Application Server Platform Edition 8, or you can get the Java 2 Platform, Enterprise Edition 1.4 SDK bundle. This bundle contains:

- J2SE 1.4.2_04 platform

- Java 2 Platform, Enterprise Edition 1.4 Samples (contains BluePrints and J2EE 1.4 SDK samples)

- Sun Java System Application Server

You can also get the Java 2 Platform, Enterprise Edition 1.4 Samples bundle, which contains just the BluePrints and J2EE 1.4 SDK samples.

Key Features

The Sun Java System Application Server includes the following components:

- Application Server

- Sun Java™ System Message Queue

- PointBase Database Server and Type 4 JDBC Driver

- Java 2 Software Development Kit, Standard Edition

- Support for Foreign JMS Providers

Application Server

The Application Server supports the current drafts of the J2EE 1.4 specifications.

- Java Web Services APIs:

    ○ Java API for XML Processing (JAXP) version 1.2

    ○ Java API for XML-based Remote Procedure Calls (JAX-RPC) version 1.1

    ○ Java API for XML Registries (JAXR) 1.0

    ○ SOAP with Attachments API for Java (SAAJ) version 1.2

- Enterprise JavaBeans specification version 2.1

- WS-I Basic Profile specification 1.0

- J2EE Deployment API version 1.1

- J2EE Management specification version 1.0

- Java Authorization Contract For Containers (JACC) 1.0

- JavaServer Pages specification version 2.0

- Java Servlet specification version 2.4

- Java Naming and Directory Interface (JNDI) specification version 1.2

- Java Database Connectivity (JDBC) specification version 3.0

- JavaMail API version 1.3

- JavaBeans Activation Framework (JAF) version 1.0.

- J2EE Connector specification version 1.5

- JavaServer Pages Standard Tag Library (JSTL) version 1.1

Additionally, the Application Server supports the following J2EE 1.4 features:

- Includes runtime control, process, and thread management

- Embeds a highly scalable and high-performance Container Managed Persistence runtime

- Java Transaction Service-based, all-Java transaction manager

- Object Request Broker (ORB)

- J2EE verifier utilities

- Customized Apache Ant tasks (asant utility)

- Browser-based administration tool (Admin Console)

- Command-line administration tool (asadmin) with interactive and scripting modes

- Graphical deployment utility (deploytool) to simplify application, enterprise bean, and connector packaging and deployment

- Sample applications that demonstrate the features of the Application Server and J2EE technology

Sun Java™ System Message Queue

Also available as a separate product, the Sun Java System Message Queue product is an integral part of the Application Server. This all-Java programming language component delivers a robust JMS provider for both JMS clients and message-driven beans. The Sun Java System Message Queue is integrated with the Sun Java System Application Server Platform Edition 8 through a resource adaptor that is based on J2EE Connector Architecture.

The Sun Java System Application Server Platform Edition 8 product includes the Sun Java System Message Queue 3.5 Service Pack 1 Platform Edition. To get enhanced JMS features, the Sun Java System Message Queue Enterprise Edition can be purchased separately.

PointBase Database Server and Type 4 JDBC Driver

PointBase Server 4.8 relational database is included with the Sun Java System Application Server 8 Platform Edition to support the sample applications and development of JDBC-based applications. PointBase's Type 4 driver supporting the JDBC™ API ("JDBC driver") is preconfigured during installation. Tables are also created and populated for all of the sample applications that depend on JDBC. The bundled PointBase distribution has a 5 MB total database size limitation.

Java 2 Software Development Kit, Standard Edition

The Sun Java System Application Server is certified with the JDK 1.4.2_04 platform.

Support for Foreign JMS Providers

Addendum to the Sun Java System Application Server 8 Developer's Guide.

Sun Java System Application Server supports foreign JMS providers for message outflow and inflow. Thus an EJB component or servlet can send messages to a JMS provider other than Sun Java System Message Queue. A message-driven bean can also receive messages from a foreign JMS provider.

Foreign JMS provider support is part of the Connector 1.5 specification requirements. Each JMS provider must provide a resource adapter, which can be deployed and configured as described here:

http://developers.sun.com/prodtech/appserver/reference/techart/as8_connectors

Each JMS provider supplies an administration tool, which must be used for any provider-specific configuration.

System Requirements

This section lists the requirements that must be met before installing the Sun Java System Application Server Platform Edition 8 product.

- Platform Requirements
- JDBC Drivers and Databases
- Other Requirements

Platform Requirements

The following table lists the operating systems that are supported for Sun Java System Application Server product.

Operating System	Architecture	Minimum Memory	Recommended Memory	Minimum Disk Space	Recommended Disk Space	JVM
Sun Solaris 8, 9 (SPARC)	32 and 64 bit	256 MB	512 MB	250 MB free	500 MB free	JDK 1.4.2_04
Solaris 9 (x86)	32 and 64 bit	256 MB	512 MB	250 MB free	500 MB free	JDK 1.4.2_04
Sun Java Desktop System	32 bit	256MB	512MB	83 MB free	300 MB free	JDK 1.4.2_04
Redhat Enterprise Linux 2.1 and 3.0	32 bit	256 MB	512 MB	83 MB free	300 MB free	JDK 1.4.2_04

Operating System	Architecture	Minimum Memory	Recommended Memory	Minimum Disk Space	Recommended Disk Space	JVM
Microsoft Windows 2000 Professional Service Pack 4+	Intel 32 bit	256 MB	512 MB	250 MB free	500 MB free	JDK 1.4.2_04
Microsoft Windows XP Professional Service Pack 1+	Intel 32 bit	256 MB	512 MB	250 MB free	500 MB free	JDK 1.4.2_04
Microsoft Windows 2000 Server SP4+	Intel 32 bit	256 MB	512 MB	250 MB free	500 MB free	JDK 1.4.2_04
Microsoft Windows Advanced Server Service Pack 4+ , Microsoft Windows Server 2003	Intel 32 bit	256 MB	512 MB	250 MB free	500 MB free	JDK 1.4.2_04

On UNIX, you can check your operating system version using the `uname` command. Disk space can be checked using the `df` command.

Solaris Patch Requirements

It is recommended that Solaris 8 users have the "Sun recommended patch cluster" installed. This patch cluster is available under "Recommended and Security Patches" here:

```
http://sunsolve.sun.com/
```

JDBC Drivers and Databases

The Sun Java System Application Server is designed to support connectivity to any DBMS with a corresponding JDBC driver. For a list of components that Sun has tested and found to be acceptable for constructing J2EE compatible database configurations, please refer to the following table:

Table 2   J2EE Compatible JDBC Drivers

JDBC Vendor	JDBC Driver Type	Supported Database Server
i-net Software	Type 4	Oracle (R) 9.2, 8.17
i-net Software	Type 4	Sybase ASE 12.5.1
i-net Software	Type 4	MS SQL Server 2000 Service Pack 1
IBM	Type 2	IBM DB2 8.1 Service Pack 3

Table 2   J2EE Compatible JDBC Drivers

JDBC Vendor	JDBC Driver Type	Supported Database Server
PointBase	Type 4	PointBase Network Server 4.8

For more information about i-net Software, see:

http://www.inetsoftware.de/

The following table identifies additional supported JDBC drivers; however these drivers are not J2EE compatible.

Table 3   JDBC Drivers not J2EE compatible

JDBC Vendor	JDBC Driver Type	Supported Database Server
Oracle	Type 4	Oracle (R) 9.2
Sybase	jConnector	Sybase ASE 12.5.1

Additional drivers have been tested to meet the JDBC requirements of the J2EE 1.4 platform with the JDBC Driver Certification Program. These drivers can be used for JDBC connectivity with the Sun Java System Application Server. While Sun offers no product support for these drivers, we will support the use of these drivers with the Sun Java System Application Server.

Other Requirements

The following additional requirements should be met before installing the Sun Java System Application Server software.

- **Free space:** your temporary directory must have a minimum of 35MB free for Sun Java System Application Server installation, and 250 MB of free space for the SDK installation.

- **Using the uninstall program:** If you need to remove the application server from your system, it is important to use the uninstall program that is included with the software. If you attempt to use another method, problems will arise when you try to reinstall the same version, or when you install a new version.

- **Free ports:** You must have seven unused ports available.

    ○ The installation program automatically detects ports in use and suggests currently unused ports for the default settings. By default, the initial default ports are 8080 for the HTTP server, and 4848 for the Admin Server.

○ The installation program will detect used ports and assign two others for you: Sun Java™ System Message Queue (by default, 7676), and IIOP (by default, 3700 for IIOP and 1060 and 1061 for IIOP/SSL). If these default port numbers are in use, the installation program will assign a random port number from the dynamic port range (note that this may not be the next available port number).

- **Starting previously-installed servers** (UNIX) — unless you are replacing the previously installed server, you should start it before you begin the Sun Java System Application Server installation process. This allows the installation program to detect ports that are in use and avoid assigning them for other uses.

- **Replacing previously-installed servers** (UNIX) — if you have an older version on the Sun Java System Application Server installed that you wish to replace with the current Application Server, you should stop it before installing the new server. Use the installation program upgrade wizard to upgrade the server.

- **Shutting down firewall** (Microsoft Windows) — You must stop any firewall software before installing the Sun Java System Application Server software, because some of this software disables all ports by default. The installation program must be able to accurately determine which ports are available.

For further compatibility information, see the Compatibility Guide available at:

```
http://docs.sun.com/db/doc/817-6084
```

Documentation

In addition to these release notes, the Application Server product includes an entire set of documentation that can be found at this location:

```
http://docs.sun.com/db/prod/s1appsrv#hic
```

The following list provides a brief description of the documentation in the Application Server collection:

- *Quick Start Guide*—Describes how to get started with the Application Server product. Focuses on initial developer exposure; is also suited for users evaluating the product.

- *Installation Guide*—Provides instructions for installing or uninstalling the Application Server software and its components. Supported platforms are listed.

- J2EE 1.4 Tutorial—Describes how to develop software using the technologies provided in the Application Server product. You can find the J2EE 1.4 Tutorial here:

```
http://java.sun.com/j2ee/1.4/docs/tutorial/doc/index.html
```

- *Compatibility Guide*—Provides information on backward compatibility issues.

- *Utility Reference*—Documentation for all command-line interface commands and the Application Server utility commands.

- Online help for the Admin Console—Provides content-specific online help for the Application Server Administration Console web-based interface.

- Online help for the `deploytool` utility—Provides content-specific online help for the Application Server `deploytool` utility.

- Online help for the `Migrationtool` utility—Provides content-specific online help for the Application Server `Migrationtool` utility. The Migration Tool is available for a separate download.

- *Troubleshooting Guide*—Provides information on solving problems associated with the Application Server product.

Known Problems and Limitations

This section describes known problems and associated workarounds for the Sun Java System Application Server 8 product. If a summary statement does not specify a particular platform, the problem applies to all platforms.

ID	Summary
none	**A slight behavioral change in the client stubs generation.**
	The client stubs are generated during deployment process. The deploy operation does not return until the client subs are fully generated.
	For performance concerns, the client stubs generation has been modified to be an asynchronous operation within deployment when the retrieval of client jars is not explicitly requested by the user; the client stubs are no longer guaranteed to be ready for use upon return of the deployment operation.
	Solution
	Do not copy any client jars directly out of the repository post deployment. Instead, the following ways are the recommended ways to retrieve client stubs with guaranteed integrity:
	Command-line interface:
	1. asadmin deploy --retrieve=true
	2. asadmin get-client-stubs
	Deploytool:
	1. Specify retrieval of client jars on the deploy screen.
	2. Check the client jar retrieval screen for any deployed applications.
4859661	**On Microsoft Windows platforms, failure to create default server domain will not always be reported as installation failure.**
	If the default server domain creation fails, failure is not always correctly logged and reported by the installer. As a result, the installation is reported as successful but attempts to start the server will produce the following error message: CLI156 Could not start the domain doman1.
	Checking the content of *install_dir*\domains\domain1 folder reveals that only docroot folder has been created instead of a full domain configuration directory structure.
	Solution
	Default server domain can be created manually. In order to recreate the domain, the *install_dir*\domains\domain1 folder should be deleted and the following command should be run:
	install_dir\bin\asadmin create-domain --adminport *admin_port* --adminuser *admin_user* --adminpassword *admin_password* --instanceport *http_port* domain1
	Values for *admin_port*, *admin_user*, *admin_password,* and *http_port* should match the values that were originally supplied at product installation.
4925544	**On Microsoft Windows, no error is reported for a silent installation where the user has no JDK installed.**
	A user is passed a previously made savestate file with the application server only executable bundle. This bundle does not contain a JDK and therefore will fail if there is no JDK installed on the machine. No error message is displayed and no log is created to record the failure.
	Solution
	Microsoft Windows users must have a minimum JDK 1.4.1_07 installed.

ID	Summary
4941546	**Misleading information for CLI start-appserv manpage.**
	Solution
	The DESCRIPTION is incorrect. No operands are required for the `start-appserv` command. Therefore, there is no point in making any references to an operand in the description.
	Since `start-appserv` starts all domains present in the given directory, `--domaindir` option should mention domains in plural.
	Sometimes "default install_dir/domains directory" is used and sometimes "default domains directory" is used in the same manpage. They both refer to the same thing.
4941547	**Misleading information for CLI stop-appserv manpage.**
	Solution
	The DESCRIPTION is incorrect. No operands are required for the `stop-appserv` command. Therefore, there is no point in making any references to an operand in the description.
	Since `stop-appserv` stops all domains present in the given directory, `--domaindir` option should read: "`--domaindir`: directory where the domains are to be stopped. If specified, path must be accessible in the filesystem. If not specified, the domain is stopped in the default `install_dir/domains` directory."
	Sometimes "`default install_dir/domains` directory" is used and sometimes "default domains directory" is used in the same manpage. They both refer to the same thing.
4942748	**Command output in EXAMPLES section of some asadmin manpages is incorrect.**
	Solution
	In the EXAMPLE section of the manpages, the output for `asadmin` commands should read: "Command *command_name* executed successfully."
4943028	**The EXAMPLE section for the list-virtual-servers command in the manpage is missing server and _asadmin.**
	Solution
	The correct EXAMPLE for the `list-virtual-servers` command is as follows:
	`./asadmin list-virtual-servers --user admin --password adminadmin --port 4848 --host localhost`
	`server`
	`_asadmin`
	Command `list-virtual-servers` executed successfully.

ID	Summary
4954870	**Newly created HTTP Listeners are never started.**

If you create an HTTP Listener, using the asadmin command or the Admin Console, with a new port number, after restarting the server, you are unable to connect to the port number of the newly generated HTTP Listener at `http://your_server:new_port_number`

Solution

Add the name (id) of the newly created HTTP Listener to the `http-listeners` attribute of the Virtual Server that is referenced by the new HTTP Listener as its Default Virtual Server.

For example, if you add a new HTTP Listener named *http-listener-3* that uses the virtual server *server* as its Default Virtual Server, follow these steps:

1. Direct your browser to the admin console, at `http://your_server:your_admin_port`
2. Login using your administrator user/password.
3. Expand the *HTTP Service node* in the left column.
4. Expand the *Virtual Servers* node under *HTTP Service.*
5. Click on the Virtual Server named *server*.
6. In the right panel, update the HTTP Listeners field by adding `http-listener-3` (separated by comma) to the list of HTTP Listeners.

| 4954884 | **Misleading information for asadmin create-domain manpage.** |

Solution

The `--passwordfile` option should state the valid values as follows:

```
http.ssl.port=1043:orb.ssl.port=1060:orb.mutualauth.port=1061:jms.port=7676:orb.listene
r.port=3700.
```

| 4971450 | **When creating new virtual server, server startup failure occurs and server throws NullPointer Exception during server startup.** |

Solution

1. While creating the virtual servers make sure the http listener list is exclusive (i.e. no other virtual server should have the same http listeners in its list).
2. While creating virtual servers and http listeners use the following sequence of `asadmin` commands

   • Create virtual server without specifying the http listener list (`httplisteners` is an optional option for `create-virtual-server` command).

   • While creating the http listener use the `defaultvs` option to specify the virtual server.

ID	Summary
4974713	Section 5.3.4.2 of the Connectors 1.5 Specification states that "the application server calls the `stop` method on the `ResourceAdapter` JavaBean to notify the resource adapter instance to stop functioning so that it can be safely unloaded. This is a shutdown notification from the application server, and this method is called by an application server thread."
	For this release, the `ResourceAdapter` JavaBean's `stop()` method is not called during application server shutdown. The `stop()` method is called only when the connector module is undeployed.
	Solution
	Add a lifecycle listener to listen for application server shutdown and disable the resource adapter module. The disable action calls the `RA.stop()`.
	For more information, see http://developer.java.sun.com/developer/bugParade/bugs/4974713.html
4997380	**User may observe a stack trace when shutting down Pointbase.**
	By executing the stopserver.sh/stopserver.bat script on any platform, the server instance of the Pointbase runtime is successfully terminated. However, due to runtime regression introduced in version 4.8, a ClassCastException is produced as follows:
| | ```
[qouyang@putian serveroption]$ ./stopserver.sh java.lang.ClassCastException
     at com.pointbase.net.netJDBCPreparedStatement.execute(Unknown Source)
     at com.pointbase.tools.toolsCommander.a(Unknown Source)
     at com.pointbase.tools.toolsCommander.go(Unknown Source)
     at com.pointbase.tools.toolsCommander.main(Unknown Source)
``` |
	Solution
	No work-around is currently available, however there is no functional regression as the script still succeeds in terminating the server process. A priority request has been logged with Pointbase to obtain a fix.
5002049	**Manpages are incorrect for the create-jvm-options and update-connector-security-map.**
	Solution
	1. The EXAMPLE for the `create-jvm-options` command should read:
	asadmin> create-jvm-options --user admin --password adminadmin --host localhost --port 4848 --profiler=false -- "-DDebug=true":"-Xmx256m":"-Dcom.sun.aas.imqBin"="\export\as8pe\imq\bin"
	Command create-jvm-options executed successfully
	2. The SYNOPSIS for the `update-connector-security-map` command should read:
	update-connector-security-map --user admin_user [--password admin_password] [--host localhost] [--port 4848] [--secure|-s] [--passwordfile filename] [--terse=false] [--echo=false] [--interactive=true] --poolname connector_connection_pool_name [--addprincipals principal-name[, principal-name]*]| [--addusergroups user-group[, user-group]*] [--removeprincipals principal-name[, prinicpal-name]*] [--removeusergroups user-group[, user-group]*] [--mappedusername user_name] [--mappedpassword password] mapname

ID	Summary
5002174	**On Microsoft Windows, intermittent corruption of PATH environment variable content after product uninstallation.**

If the "Add bin directory to PATH" installation option is selected during installation, content of PATH environment variable may get corrupted after installation.

The corrupted PATH variable will have a random set of characters appended as the last entry in the PATH. This additional PATH entry generally will not affect system functioning.

Solution

Use the Windows Control Panel, System, Advanced, Environment variables tool to edit the content of the PATH environment variable and remove the superfluous characters.

| **5003356** | **Missing entries in server.policy file after upgrade.** |

User chooses to upgrade from Sun ONE Application Server 7.x Platform Edition to Sun Java System Application Server Platform Edition 8. After the upgrade process is complete, the updated server.policy file looses two entries that were existing prior to the upgrade.

Solution

Edit *install_dir*/domains/*domain_name*/config/server.policy file as follows:

1. Add the following lines to the end of the file:

```
//ANT classes get all permissions by default
grant codeBase "file:${com.sun.aas.antLib}/-" {
  permission java.security.AllPermission;
};
```

2. Locate "// Basic set of required permissions granted..." block in the file. Add the following lines inside this grant block:

```
// work-around for pointbase bug 4864405
permission java.io.FilePermission "${com.sun.aas.instanceRoot}${/}-", "delete";
```

The original server.policy file, before upgrading, is saved in the same directory with file named server.policy.back. Please refer to this file to identify where the missing lines were placed.

| **5004719** | **Network addresses any, ANY, and INADDR_ANY are not supported.** |

When creating an HTTP listener via the Admin Console, the description of the "Network Address" field mentions any, ANY, and INADDR_ANY as valid network addresses. However, when specifying any of these values, the following error is logged when starting the application server: unable to resolve host name:any.

Solution

When creating an HTTP listener using the Admin Console, avoid using any, ANY, or INADDR_ANY as network address values.

ID	Summary
5004911	**Using Deploytool, unable to deploy an application with database mappings that were automatically created.**
	The problem occurs after creating database mappings and then trying to deploy the application.
	Create database mappings:
	1. Select a CMP EJB in the Files tree view.
	2. Click on the Sun-specific Settings button in the EJB JAR General tab to display the CMP Database dialog.
	3. Click on the Create Database Mappings button to display the dialog.
	4. In the Create Database Mappings dialog, select Automatically Generate Necessary Tables.
	5. Select OK.
	6. Close the CMP Database dialog.
	Deploy the application:
	1. Select the application in the Files tree view.
	2. Select Deploy from the Tools menu.
	The application is not deployed and the following error is displayed:
	Referenced schema is not present in jar: RosterAppJ2DB_team-ejb Adding the schema and saving the archive or replacing the mapping, by triggering the Create Database Mapping action, will correct this.
	Solution
	Dismiss the error dialog.
	Save the application:
	1. Select the application in the Files tree view.
	2. Select Save from the File menu.
	Deploy the application:
	1. Select the application from the Files tree view.
	2. Select Deploy from the Tools menu.

ID	Summary
5005059	**Using Deploytool, when two CMP EAR files are opened with the same name, the mapping is not displayed for the second EAR.**
	To view the mapping, select the Sun-specific Settings button in the EJB JAR General tab to display the CMP Database dialog. A Packaging Warning message is displayed: Referenced schema not present in jar ... No mapping is displayed for the second EAR opened.
	If you attempt to perform the mapping again, another error message is displayed: Cannot create schema from cmproster.dbschema. Please check the schema file ...
	Solution
	Close the first application and remove *earname.dir* under the deploytool temporary directory. The temporary directory location is displayed on the General panel of the Edit Preferences dialog. Select Preferences under the Edit menu to display the Edit Preferences dialog.
5005168	**On Microsoft Windows, while using Deploytool, after adding a CMP EJB to an EAR file, it may not be possible to create database mappings for that EJB.**
	When adding a CMP EJB that has a dbschema file to an EAR, an exception is thrown and the mappings dialog cannot be closed. Deploytool must be terminated externally.
	Add CMP EJB that has a dbschema file to an EAR:
	1. Select Add to Application --> Enterprise JavaBean JAR from the File menu.
	Create database mappings:
	1. Click on the Sun-specific Settings button on the EJB JAR General tab to bring up the CMP Database dialog.
	2. Click on the Create Database Mappings button.
	3. Select Map to Tables in Database Schema File and choose the schema file.
	When trying to create database mappings, an exception is thrown and the mappings dialog cannot be closed: javax.ejb.EJBException: nested exception is: java.lang.ClassNotFoundException
	Solution
	After adding the CMP EJB to an EAR, save the application:
	1. Select the application in the Files tree view.
	2. Select Save from the File menu.
	Restart deploytool:
	1. Select Exit from the File menu.
	2. Start deploytool.
	Create database mappings:
	1. Click on the Sun-specific Settings button on the EJB JAR General tab to bring up the CMP Database dialog.
	2. Click on the Create Database Mappings button.
	3. Select Map to Tables in Database Schema File and choose the schema file.

ID	Summary
5005169	**Using Deploytool, cannot Create Database Mappings from a dbschema file that has been added to a CMP EJB which is embedded in an EAR and already contains a dbschema file.**
	The problem happens when a second dbschema file is added to a CMP EJB which is embedded in an EAR and already contains a dbschema file. The database mappings cannot be created from the second dbschema file.
	Solution
	1. Save the application.
	2. Restart the deploytool.
	3. Create database mappings using the newly added dbschema file:
	•Click on the Sun-specific Settings button on the EJB JAR General tab to display the CMP Database dialog.
	•Click on the Create Database Mappings button.
	•Select Map to Tables in Database Schema File and choose the newly added schema file.
	•Click OK.
5005792	**Unsupported: RedHat Fedora (core 1) requires the libstdc++ package to be installed in order to install J2EE1.4SDK.**
	Solution
	The install package named `compat-libstdc++-7.3-2.96.118.i386.rpm` can be found on disk 3 of the Fedora ISO installation CD image for corel.

ID	Summary
5006854	**Deploying to a virtual server using asadmin command fails if the virtual server does not have at least one listener for clear HTTP and at at least one listener for HTTP over SSL.**

```
#|2004-03-02T18:08:48.588-0500|WARNING|sun-appserver-pe8.0|javax.enterprise.system.
tools.admin|_ThreadID=11;|ADM1026:Redeployment failed - Detailed Message: java.lang.NullPointerException
at
com.sun.enterprise.admin.mbeans.ApplicationsConfigMBean.getHostName(ApplicationsConfigMBean.java:3045)
at
com.sun.enterprise.admin.mbeans.ApplicationsConfigMBean.setHostAndPort(ApplicationsConfigMBean.java:303
5)
at com.sun.enterprise.admin.mbeans.ApplicationsConfigMBean.deploy(ApplicationsConfigMBean.java:274)
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
```

Solution

Deploy without specifying a virtual server then change the virtual server using the `asadmin set` command:

```
asadmin deploy hello.ear
asadmin set server.application-ref.hello.virtual-servers=newvs
```

This will make the web module(s) contained within hello.ear available only on newvs.

This solution does not work completely for web archives (war file). After running the suggested commands, the web archive's context root will be available only on the specified virtual server till the server is running.

However, a restart of the server will make the web archive's context root available on all virtual servers. To workaround this problem, please package your web modules within an application archive (ear file) for deployment.

| 5014416 | **Supporting unknown Primary Key in CMP beans.** |

Solution

To use unknown Primary Key feature for a CMP bean, do the following:

1. Do not define primary key fields in your bean.

2. Define primary key class as `java.lang.Object` in deployment descriptor.

If you map your CMP bean to an existing table, make sure the table:

- has only one primary key column

- is not mapped to any other CMP field

- is a NUMERIC type with precision 19 or higher (to store a long value)

If you rely on the CMP module to generate the mapping, you do not need to perform any other steps. When such a bean is created, the primary key value will be populated by the Persistence Manager.

ID	Summary
5014452	**Addendum to the Sun Java System Application Server 8 Developer's Guide.**
	Solution
	Added support for CMP beans with `java.lang.String` primary key fields mapped to a fixed length character.
5014466	**Addendum to the Sun Java System Application Server 8 Developer's Guide.**
	Solution
	Supporting join table to map all types of CMP relationships:
	CMR field that represents 1-N (0r 1-1) relationship between CMP beans can now be mapped to join table.
	For example:
	• Department bean is mapped to a DEPARTMENT table with a primary key column "deptid".
	• Employee bean is mapped to an EMPLOYEE table with primary key column "empid".
	• CMR field "employees" in a Department bean is a 1-N relationship to an Employee bean, and represents a Collection of employees in this department.
	• This field can be mapped:
	• Either by a foreign key "deptid" field in an EMPLOYEE table
	• Or a join table EMPLOYEE_DEPARTMENT with columns "empid" and "deptid" that are both foreign keys to the corresponding columns in EMPLOYEE and DEPARTMENT tables. you might consider a unique constraint on the "empid" column to guarantee that each Employee belongs to only one department.
5015099	**deploydir instruction in QuickStart.html incorrect.**
	Solution
	The URL for accessing the deployed quickstart hello application using deploydir should be: "http://localhost:8080/hello"

How to Report Problems

Use the following resources to handle problems you may encounter with the Application Server product:

- Feedback Submittal form—A form for submitting feedback on the Application Server product, here:

  `http://java.sun.com/docs/forms/J2EE14SubmittalForm.html`

- J2EE-INTEREST list—A mailing list for J2EE questions, here:

```
http://archives.java.sun.com/archives/j2ee-interest.html
```

• Bug database on Java Developer Connection—To view bugs or to submit a bug, use the Java Developer Connection Bug Parade here:

```
http://developer.java.sun.com/servlet/SessionServlet?url=/developer/bugPar
ade/index.jshtml
```

• Java Technology Forums—An interactive message board for sharing knowledge and questions about Java technologies and programming techniques. Use the J2EE SDK forum here for discussions related to the Sun Java System Application Server 8 Platform Edition product:

```
http://forum.java.sun.com/
```

For More Information

Useful information can be found at the following locations:

• Application Server product information:

http://wwws.sun.com/software/products/appsrvr_pe/index.html

• Java developer resources:

```
http://developer.java.sun.com/
```

• Java 2 Platform, Enterprise Edition (J2EE) site:

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

• Application Server product documentation:

```
http://docs.sun.com/db/prod/s1appsrv#hic/
```

• Sun Microsystems product documentation:

```
http://docs.sun.com/
```

Revision History

This section lists the changes that have been made in these release notes after the initial release of the Application Server product

Revision Date	Description of Change
March 2004	Sun Java System Application Server 8 release.