Sun Java System Application Server Platform Edition 9 Release Notes

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

Part No: 819–3653–11
May 2008

080516@20209

Contents

◆ ◆ ◆ CHAPTER 1

Overview

The Sun Java™ System **Application Server Platform Edition 9** product is a Java EE 5 platform-compatible server for the development and deployment of Java EE applications and Java Web Services. Production use of this server is free of charge. Sun Java System Application Server Platform Edition is free for development, deployment and redistribution. Customers interested in redistribution should contact Sun OEM sales (`http://www.sun.com/software/products/appsrvr/appsrvr_oem.html`) for a redistribution license.

This document contains the following sections:

About These Notes

These Release Notes contain important information available at the time of release of Sun Java System Application Server Platform Edition 9. New features and enhancements, known issues and limitations, and other information are addressed here. Read this document before you begin using Application Server 9.

The most up-to-date version of these release notes can be found at the Sun documentation Web site (`http://docs.sun.com/app/docs/coll/1343.3`). Check the Web site prior to installing and setting up your software and then periodically thereafter to view the most up-to-date release notes and product documentation.

Third-party URLs are referenced in this document and provide additional, related information.

Note – Sun is not responsible for the availability of third-party Web sites mentioned in this document. Sun does not endorse and is not responsible or liable for any content, advertising, products, or other materials that are available on or through such sites or resources. Sun will not be responsible or liable for any actual or alleged damage or loss caused by or in connection with the use of or reliance on any such content, goods, or services that are available on or through such sites or resources.

Release Notes Revision History

This section lists the revision history of these release notes.

TABLE 1–1 Release Notes Revision History

Revision Date	Description
May 2007	FCS release of version 9.0 PE.
May 2008	Added "System Virtualization Support" section.

Accessibility Features

To obtain accessibility features that have been released since the publishing of this media, consult Section 508 product assessments available from Sun upon request to determine which versions are best suited for deploying accessible solutions. Updated versions of applications can be found at: `http://sun.com/software/javaenterprisesystem/get.html`

For information on Sun's commitment to accessibility, visit `http://sun.com/access`.

Related Documentation

In addition to these Release Notes, the Application Server product includes an entire set of documentation (`http://docs.sun.com/app/docs/prod/sjs.asse#hic`).

The following table summarizes the books included in the Application Server Platform Edition core application documentation set.

TABLE 1–2 Books in This Documentation Set

Book Title	Description
Sun Java System Application Server Platform Edition 9 Documentation Center	**One stop location to access all Application Server topics.**
Sun Java System Application Server Platform Edition 9 Quick Start Guide	How to get started with the Sun Java System Application Server product.

TABLE 1–2 Books in This Documentation Set *(Continued)*

Book Title	Description
Sun Java System Application Server Platform Edition 9 Documentation Center	**One stop location to access all Application Server topics.**
Sun Java System Application Server Platform Edition 9 Installation Guide	Installing the Sun Java System Application Server software and its components.
Sun Java System Application Server Platform Edition 9 Application Deployment Guide	Information assembling and deploying Java EE applications on the Sun Java System Application Server.
Sun Java System Application Server Platform Edition 9 Developer's Guide	Creating and implementing Java 2 Platform, Enterprise Edition (Java EE™ platform) applications intended to run on the Sun Java System Application Server that follow the open Java standards model for Java EE components and APIs. Includes general information about developer tools, security, assembly, deployment, debugging, and creating lifecycle modules.
Sun Java System Application Server Platform Edition 9 Java EE Tutorial	Using Java EE 5 platform technologies and APIs to develop Java EE applications and deploying the applications on the Sun Java System Application Server.
Sun Java System Application Server Platform Edition 9 Administration Guide	Configuring, managing, and deploying the Sun Java System Application Server subsystems and components from the Administration Console.
Sun Java System Application Server Platform Edition 9 Administration Reference	Editing the Sun Java System Application Server configuration file, `domain.xml`.
Sun Java System Application Server Platform Edition 9 Upgrade and Migration Guide	Migrating your applications to the new Sun Java System Application Server programming model, specifically from Application Server 6.*x*, 7, and 8.*x*. This guide also describes differences between adjacent product releases and configuration options that can result in incompatibility with the product specifications.
Sun Java System Application Server Platform Edition 9 Troubleshooting Guide	Solving Sun Java System Application Server problems.
Sun Java System Application Server Platform Edition 9 Error Message Reference	Solving Sun Java System Application Server error messages.
Sun Java System Application Server Platform Edition 9 Reference Manual	Utility commands available with the Sun Java System Application Server; written in manpage style. Includes the `asadmin` command line interface.

Sun Welcomes Your Comments

Sun is interested in improving its documentation and welcomes your comments and suggestions.

To share your comments, go to `http://docs.sun.com` and click Send Comments. In the online form, provide the document title and part number. The part number is a seven-digit or nine-digit number that can be found on the title page of the book or at the top of the document. For example, the title of this book is *Sun Java System Application Server Platform Edition 9 Release Notes*, and the part number is 819-3653.

Additional Sun Resources

Useful information about the Application Server product can be obtained from the following locations:

- Java EE Feedback Submittal Form (`http://java.sun.com/docs/forms/javaeesubmittalform.html`) — Submit feedback about the Sun Java System Application Server
- J2EE-INTEREST List (`http://archives.java.sun.com/archives/j2ee-interest.html`) — Mailing list for questions about the Java EE platform
- Bug Database on Java Developer Connection (`http://developer.java.sun.com/developer/bugParade/index.jshtml`) — View or submit bugs related to the Sun Java System Application Server
- Glassfish Community (`http://java.sun.com/javaee/glassfish`) — Free, open source project community through which Application Server and other implementations of the newest features in the Java EE 5 platform are developed
- Java Enterprise Community (`http://community.java.net/java-enterprise/`) — Part of java.net (`http://java.net/`) centered around Java EE applications, components, and tools
- EJB 3.0 Tech Tips (`http://java.sun.com/developer/EJTechTips/`) — Useful EJB 3.0 tech tips in the Developer area of java.sun.com (`http://java.sun.com/developer`); in particular:
  - Accessing the Bean Environment in EJB Session Beans (`http://java.sun.com/developer/EJTechTips/2005/tt0930.html#1`)
  - Converting a POJO to a Persistent Entity (`http://java.sun.com/developer/EJTechTips/2005/tt1122.html#2`)
- Sun Developer Forums (`http://developers.sun.com/forums/`) — Interactive message boards for sharing knowledge and questions about technologies and programming techniques; use the Application & Integration Servers (`http://swforum.sun.com/jive/category.jspa?categoryID=10`) forum for discussions related to the Application Server
- Java Technology Forums (`http://forum.java.sun.com/index.jspa`) — Interactive message boards for sharing information about Java technologies in general

- Java EE SDK Forums (`http://forum.java.sun.com/forum.jspa?forumID=136`) — Interactive message boards for sharing information about the Java EE SDK in particular
- Sun Software Support Service Standard (`http://www.sun.com/service/support/software/standard.html`) — Contracts are available for purchase
- Sun Developer Network (`http://developers.sun.com/user_registration/whyregister.html`) — Join to obtain further help and to keep on top of the latest Java developments
- docs.sun.com (`http://docs.sun.com/`) — Complete documentation for Application Server and other Sun products
- Sun Java System Application Server home page (`http://www.sun.com/appserver`) — Data sheets, press information, news, and other information about Application Server

◆ ◆ ◆ CHAPTER 2

About Application Server Platform Edition 9

The Sun Java™ System **Application Server Platform Edition 9** is a Java EE 5 platform-compatible server for the development and deployment of Java EE applications and Java technology-based web services.

This section includes:

What's New in the 9 Release

The Sun Java System Application Server Platform Edition 9 implements many new features:

- **Java EE 5 platform support** — Sun Java System Application Server Platform Edition 9 implements the Java EE 5 specification to deliver one of the best application runtimes for next-generation enterprise applications and web services. Application Server 9 implements the following Java EE standards:
  - Enterprise Java Beans 3.0
  - JAXB 2.0
  - Java Persistence
  - Java Server Faces 1.2
  - Java Server Pages 2.1 (JSP 2.1)
  - Java Server Pages Standard Tag Library (JSTL) 1.2
  - Streaming API for XML (StAX)
  - Web Services Metadata
  - Java API for XML based Web Services 2.0 (JAX-WS 2.0)
  - Common Annotations for the Java Platform 1.0 (CAJ 1.0)
  - Java Servlet 2.5

The complete list of Java EE 5 platform technologies is provided later in these notes, in "Java EE 5 Platform APIs" on page 23

- **Improved Developer Productivity** — Application Server 9 provides a rich developer experience with its open modular extensible architecture and lightweight, low footprint, high performance runtime. Fast application deployment makes iterative application development quick and easy. The product can be easily integrated into popular IDEs. NetBeans and Sun Java Studio Creator integrate Application Server 9, providing a superior development environment.
- **Java 2 Standard Edition 5.0 Support** — Application Server 9 supports Java 2 Standard Edition 5.0 (`http://java.sun.com/j2se/1.5.0/`), which includes enhanced management and monitoring features and many performance and scalability improvements.
- **Sophisticated Administration** — Application Server 9 provides remote secure management using a browser-based Administration Console and a scriptable command line interface. New Admin Console features include:
  - Comprehensive web services management support including web services discovery, management, monitoring, and provisioning.
  - Enhanced application monitoring, visualization, and diagnosis.
  - JMX MBean API to provide for remote secure programmatic administration and monitoring using JMX.
  - Built-in management rules and triggers that can be expanded programmatically.
- **Platform Support** — Application Server 9 supports wide range of operating systems, databases, web servers, locales, and hardware. See the "Platform Requirements" on page 16 later in these notes for the complete list.
- **64-bit Support** — Application Server 9 adds support for the 64-bit Solaris Operating System.
- **Java Web Services Developer Pack Integration** — Application Server 9 is integrated with the Java Web Services Developer Pack (`http://java.sun.com/webservices/downloads/webservicespack.html`) (Java WSDP) 2.0 to provide support for the latest web services standard--including Web Services Metadata 1.0 (specifies annotations for web services), JAX-WS 2.0 (specifies web services API for Java platform), and JAXB 2.0 (specifies Java and XML binding). In addition, it includes a Java EE Service Engine based on the Java Business Integration (JBI) standard.
- **JMS Connectivity** — Application Server 9 provides built-in connectivity with IBM MQ Series and Sun Java System Message Queue Server.
- **Long-Running Transactions** — Application Server 9 adds support for long-running distributed transactions.
- **JDBC™ Drivers** — Application Server 9 connects to any database server with a JDBC driver. For a list of components that Sun has tested and found to be acceptable for constructing Java EE compatible database configurations, see "JDBC Drivers and

Databases" on page 17 later in these notes. Application Server includes DataDirect™ JDBC drivers for major databases that can be used for deployment.

- **Java DB Database** — Application Server 9 bundles a Java DB database based on the Apache Derby database, making it possible to develop and deploy end-to-end Java EE applications.
- **Call Flow Monitoring** — Application Server 9 can be configured to monitor an incoming request as it flows through various containers in the application server, and through the user application code. For example, the Application Server can break down the time spent in the web container, web application code, EJB container, and EJB application code. The collected information is stored in a database, and is then available for query and analysis.
- **Self Management Rules** — Application Server 9 provides a powerful and flexible infrastructure to automate application server management tasks. A self-management rule consists of an event and an action. Examples of events include message logging, monitoring threshold, timer, and JMX notifications. Actions are logic defined by the user, which are then encapsulated in a JMX MBean. When an event is triggered, and it matches one of the events defined in a self management rule, the associated action will be executed. For example, an administrator might define a management rule such that he or she receives an email when a SEVERE message is logged in the application server.
- **Migration and Upgrade Tools** — Included tools enable you to verify Java EE applications for standards conformance and portability, help with migrations from other Java EE application servers (JBoss, WebLogic, WebSphere), and aid in upgrading from previous versions of Sun's Application Servers. These migration tools are also available separately at `http://java.sun.com/j2ee/tools/migration/index.html`
- **Open Source and GlassFish Community** — In June 2005, Sun launched the GlassFish (`http://java.sun.com/javaee/glassfish/`) community with the goal of developing a free, open source, commerical-grade application server that implements the newest features of the Java EE 5 platform and related enterprise technologies. Application Server Platform Edition 9 is based on the source code developed by Sun engineers and the GlassFish community.

Hardware and Software Requirements

This section lists the requirements that must be met before installing the Sun Java System Application Server Platform Edition 9 product.

Platform Requirements

The following table lists the operating systems with which the Sun Java System Application Server Platform Edition 9 is compatible.

TABLE 2–1 Supported Operating Systems

Operating System	Minimum Memory	Recommended Memory	Minimum Disk Space	Recommended Disk Space	JVM
Sun Solaris 9, 10 (SPARC) Solaris 9, 10 (x86)	512 MB	512 MB	250 MB free	500 MB free	J2SE 5.0 Java SE 6 (when available)
64–bit Sun Solaris 10 (SPARC, x86)	512 MB	512 MB	250 MB free	500 MB free	J2SE 5.0 Java SE 6 (when available)
Redhat Enterprise Linux 3.0 U1, 4.0	512 MB	1 GB	250 MB free	500 MB free	J2SE 5.0 Java SE 6 (when available)
Windows Server 2000 SP4+ Windows 2000 Advanced Server SP4+ Windows Server 2003 Windows XP Pro SP1+	1 GB	2 GB	500 MB free	1 GB free	J2SE 5.0 Java SE 6 (when available)

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

Note – You must use the NTFS file system rather than FAT or FAT32 when running the Application Server on any Microsoft Windows platform.

System Virtualization Support

System virtualization is a technology that enables multiple operating system (OS) instances to execute independently on shared hardware. Functionally, software deployed to an OS hosted in a virtualized environment is generally unaware that the underlying platform has been virtualized. Sun performs testing of its Sun Java System products on select system virtualization

and OS combinations to help validate that the Sun Java System products continue to function on properly sized and configured virtualized environments as they do on non-virtualized systems. For information about Sun support for Sun Java System products in virtualized environments, see System Virtualization Support in Sun Java System Products.

Important Patch Information

For the current list of required patches for Sun Java System Application Server Platform Edition 9 go to `http://sunsolve.sun.com` and select Patches and Updates (`http://sunsolve.sun.com/pub-cgi/show.pl?target=patchpage`). Follow the Sun Java System Application Server Platform Edition 9 links. As operating system patch requirements change and patches to Java Enterprise System components become available, updates will be made available on SunSolve, initially in the form of recommended patch clusters.

Solaris Patch Requirements

It is recommended that Solaris 9, 10 (x86, SPARC) users have the "Sun recommended patch cluster" installed. This patch cluster is available under "Recommended and Security Patches" on the SunSolve (`http://sunsolve.sun.com/pub-cgi/show.pl?target=patchpage`) Web site.

JDBC Drivers and Databases

Table 2–2 lists databases and drivers that meet the Java EE compatibility requirements. All supported configurations of the Sun Java System Application Server Platform Edition must contain at least one database/driver combination from this table, such as the bundled Java DB database and driver. In addition, the Application Server is designed to support JDBC connectivity to any additional DBMS with a corresponding JDBC driver.

TABLE 2–2 Java EE-Compatible JDBC Drivers

JDBC Driver Vendor	JDBC Driver Type	Supported Database Server
Java DB Network Client	Type 4	Apache Derby 10.1
DataDirect 3.5	Type 4	Oracle 9i, 10g

Table 2–3 lists all the additional databases and drivers Sun has tested internally.

TABLE 2–3 Additional Supported JDBC Drivers and Databases

JDBC Driver Vendor	JDBC Driver Type	Supported Database Server
DataDirect 3.5	Type 4	Sybase ASE 12.5, 15 Microsoft SQL Server 2000, 2005 IBM DB2 8.1, 8.2

TABLE 2–3 Additional Supported JDBC Drivers and Databases *(Continued)*

JDBC Driver Vendor	JDBC Driver Type	Supported Database Server
IBM	Type 2	IBM DB2 8.1, 8.2
Microsoft MS SQL	Type 4	SQLServer 2000, 2005
MySQL Connector/J Driver 3.1	Type 4	MySQL 5
Oracle	Type 4	Oracle 9i, 10g
Postgres	Type 4	Postgres 8.1
Sybase	Type 4	ASE 12.5, 15

In general, the Application Server Platform Edition 9 supports all JDBC drivers that meet the Java EE specification.

Using the Bundled Java DB Database

This section provides instructions for using the Java DB database implementation bundled with Application Server 9. Java DB is based on the Apache Derby database.

- "Starting and Stopping the Java DB Database" on page 18
- "Java DB Utility Scripts" on page 18
- "Exporting Tables from Pointbase to Java DB" on page 19

Starting and Stopping the Java DB Database

Sun Java System Application Server 9 introduces two new `asadmin` commands for starting and stopping the Java DB Network Server.

- Use the `asadmin start-database` command to start an instance of the Java DB network server:

  ```
  start-database [--dbhost 0.0.0.0] [--dbport 1527] [--dbhome path]
  ```

  The default value for the host is `0.0.0.0`, which enables Java DB to listen on `localhost` as well as the IP/hostname interfaces. The value for the `dbhome` property is the location of where the Java DB databases reside. The default `path` is *appserver_install_dir*`/javadb`.

- Use the `asadmin stop-database` command to shut down a running instance of the Java DB network server:

  ```
  stop-database [--dbhost 0.0.0.0] [--dbport 1527]
  ```

Java DB Utility Scripts

The Java DB configuration that ships with Application Server 9 also includes several useful scripts which can help you use Java DB. The following scripts are available for use in the <*appserver_install_dir*>`/javadb/frameworks/NetworkServer/bin` directory:

- `startNetworkServer.ksh/bat` — Script to start the network server
- `stopNetworkServer.ksh/bat` — Script to stop the network server
- `ij.ksh/bat` — interactive JDBC scripting tool
- `dblook.ksh/bat` — Script to view all or part of the DDL for a database
- `sysinfo.ksh/bat` — Script to display versioning info regarding the Java DB environment
- `NetworkServerControl.ksh/bat` — Script which provides a means of executing commands on the `NetworkServerControl` API

▼ To Configure Your Environment to Run the Java DB Utility Scripts

1. **Set the `DERBY_INSTALL` environment variable to point to the *<appserver_install_dir>*`/javadb` directory.**
2. **Unset your `CLASSPATH` environment variable.**
3. **You can also optionally set the following properties:**

   a. **`DERBY_SERVER_HOST` to the host on which the network server will listen.**

      Can also be set to `0.0.0.0` to enable all listeners.

   b. **`DERBY_SERVER_PORT` to the port number on which the network server will listen.**

See Also For more information about these utilities, see the Derby Tools (`http://db.apache.org/derby/docs/10.1/tools/`) and Admin (`http://db.apache.org/derby/docs/10.1/adminguide/`) guides.

Exporting Tables from Pointbase to Java DB

This example shows how to capture the DDL for a Pointbase table in commander and create the same table in Java DB using NetBeans 5.0. Another option for doing this is by using the commander tool and the `unload database` command:

```
./startcommander.sh
Do you wish to create a new Database. (Yes (Y) or No (N))? [default: N]:
Enter product to connect with: (Embedded (E) or Server (S))? [default: E]: e
Enter driver to use? [default: [com.pointbase.jdbc.jdbcUniversalDriver]:
Enter database URL? [default: [jdbc:pointbase:embedded:sample]:
Enter Username? [default: PBPUBLIC]:
Enter Password? [default: PBPUBLIC]:

PointBase Commander 5.2 ECF build 294 size restricted version EMBEDDED

Interactive SQL command language. SunOS/5.9
```

```
(C) Copyright 2004 DataMirror Mobile Solutions, Inc. All rights reserved.

Licensed to: Sun_customer_demo_use
For commercial version contact PointBase at:
pointbase.com
PHONE: 1-877-238-8798 (US & CANADA)
       1-408-961-1100 (International)
WEBSITE: www.pointbase.com

SQL>unload database sampledb.sql;
SQL> unload database sampledb.sql;
SQL> 13 Row(s) Unloaded. (PBPUBLIC.CUSTOMER_TBL)
SQL> 4 Row(s) Unloaded. (PBPUBLIC.DISCOUNT_CODE_TBL)
SQL> 30 Row(s) Unloaded. (PBPUBLIC.MANUFACTURE_TBL)
SQL> 11 Row(s) Unloaded. (PBPUBLIC.MICRO_MARKETS_TBL)
SQL> 9 Row(s) Unloaded. (PBPUBLIC.OFFICE_TBL)
SQL> 4 Row(s) Unloaded. (PBPUBLIC.OFFICE_TYPE_CODE_TBL)
SQL> 15 Row(s) Unloaded. (PBPUBLIC.ORDER_TBL)
SQL> 6 Row(s) Unloaded. (PBPUBLIC.PRODUCT_CODE_TBL)
SQL> 30 Row(s) Unloaded. (PBPUBLIC.PRODUCT_TBL)
SQL> 10 Row(s) Unloaded. (PBPUBLIC.SALES_REP_DATA_TBL)
SQL> 10 Row(s) Unloaded. (PBPUBLIC.SALES_REP_TBL)
SQL> 52 Row(s) Unloaded. (PBPUBLIC.SALES_TAX_CODE_TBL)
SQL> 12 Table(s) Unloaded.
SQL> quit;
```

The results from executing the `unload database` command is written in the above example to the file `sampledb.sql`. The `sampledb.sql` file contains all of the DDL required to create the necessary tables and indexes. It also contains the DML to insert the data back into the database. The commander command `RUN` is intended to be used import the data into another Pointbase database using the script that was generated. Here is an example of what the `INSERT` statements and associated data look like in the generated file:

```
INSERT INTO "ADVENTURE"."CATEGORY" (
"CATID", "LOCALE", "NAME", "DESCRIPTION", "IMAGEURI" )
VALUES( ?, ?, ?, ?, ? );
{
'ISLAND              ','en_US','Island Adventures','Experience an island /
paradise in a way fit for your needs.','Island_Adventures.gif'
'JUNGLE              ','en_US','Jungle Adventures','Experience a jungle /
paradise in a way fit for your needs.','Jungle_Adventures.gif'
'MOUNTAIN            ','en_US','Mountain Adventures','Experience an /
elevated paradise with a view.','Mountain_Adventures.gif'
'ORBITAL             ','en_US','Orbital Adventures','Experience a vacuum /
paradise with a beautiful view and where no one can hear you scream.', /
'Space_Adventures.gif'
'WESTERN             ','en_US','Western Adventures','Enjoy the Wild West. /
```

```
','Western_Adventures.gif'
'SOUTH_POLE          ','en_US','South Pole Adventures','Experience a /
frozen paradise in a way fit for your needs.','SouthPole_Adventures.gif'
};
```

You could easily edit the file generated from the commander `unload database` command so that it only consisted of the DDL (for example, it would not be hard to write a program which would process the `insert` statements). As a simple test, we use the unload database command against the Pointbase `sample` database, and then edit the generated script, making the following changes:

- Removed the phrase `Organization Heap` from the end of all `CREATE Table` statements
- Removed the `COMMIT` command
- Changed the Boolean `datatype` to be `smallint`
- Removed all of the `INSERT` statements and associated data

Next, a simple Ant script is used to execute the DDL using the `sql` target. Finally, the same experiment is repeated for the `sun-appserv-samples` database requiring the following additional changes to the generated SQL file:

- Make all changes as described above for the sample database
- Remove the `create user` commands
- Remove the `SET PATH` commands
- Change the `Decimal` precision from `38` to `max` of `31`
- Change the `float` precision from `64` to `max` of `52`
- The `SPECIFIC` keyword for `CREATE PROCEDURE` is not currently supported
- Removed the `GRANT` commands

Converting Pointbase Java procedures to work with Java DB requires some changes to the Java code as well as to the `CREATE PROCEDURE` statements. Information on creating Java DB Java procedures can be found in the Derby Reference manual (`http://db.apache.org/derby/docs/10.1/ref/`). Support for the `Boolean` datatype should be in the next release of Java DB.

Browsers

This section lists the browsers that are supported with the Sun Java System Application Server Platform Edition 9 administration console and Quick Start Guide. The browsers supported when running applications on the Application Server depend on the applications being run.

TABLE 2–4 Browsers Supported

Browser	Version
Mozilla	1.4, 1.5, 1.6, 1.7.*x*

TABLE 2–4 Browsers Supported *(Continued)*

Browser	Version
Netscape Navigator	6.2, 7.0
Internet Explorer	5.5 Service Pack 2, 6.0
Firefox	1.*x*

Upgrading the Sun Java System Application Server

Refer to the *Sun Java System Application Server Platform Edition 9 Upgrade and Migration Guide* for complete instructions for upgrading from a previous version of the Application Server to the Sun Java System Application Server Platform Edition 9.

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 200 MB 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 HTTP, `8181` for HTTPS, and `4848` for the Administration Server.
  - The installation program will detect used ports and assign two others for you: Sun JavaTM System Message Queue (by default, `7676`), and IIOP (by default, `3700` for IIOP and `3820` and `3890` 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 9 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.

- **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 *Sun Java System Application Server Platform Edition 9 Upgrade and Migration Guide*.

Java EE 5 Platform APIs

The Sun Java System Application Server Platform Edition 9 supports the Java EE 5 platform. The following table lists the enhanced APIs available on the Java EE 5 platform.

TABLE 2–5 Major API changes on the Java EE 5 Platform

API	JSR
Java EE 5	
Java Platform, Enterprise Edition 5 (http://java.sun.com/javaee/5/)	JSR 244 (http://jcp.org/aboutJava/communityprocess/pr/jsr244/)
Web Services Technologies	
Implementing Enterprise Web Services	JSR 109 (http://jcp.org/en/jsr/detail?id=109)
Java API for XML-Based Web Services (JAX-WS) 2.0	JSR 224 (http://jcp.org/en/jsr/detail?id=224)
Java API for XML-Based RPC (JAX-RPC) 1.1	JSR 101 (http://jcp.org/en/jsr/detail?id=101)
Java Architecture for XML Binding (JAXB) 2.0 (http://java.sun.com/webservices/jaxb/)	JSR 222 (http://jcp.org/en/jsr/detail?id=222)
SOAP with Attachments API for Java (SAAJ)	JSR 67 (http://jcp.org/en/jsr/detail?id=67)
Streaming API for XML (http://java.sun.com/webservices/docs/1.6/tutorial/doc/SJSXP.html)	JSR 173 (http://jcp.org/en/jsr/detail?id=173)
Web Service Metadata for the Java Platform	JSR 181 (http://jcp.org/en/jsr/detail?id=181)
Component Model Technologies	
Enterprise JavaBeans 3.0 (http://java.sun.com/products/ejb/)	JSR 220 (http://jcp.org/en/jsr/detail?id=220)
J2EE Connector Architecture 1.5 (http://java.sun.com/j2ee/connector/)	JSR 112 (http://jcp.org/en/jsr/detail?id=112)
Java Servlet 2.5 (http://java.sun.com/products/servlet/)	JSR 154 (http://jcp.org/en/jsr/detail?id=154)
JavaServer Faces 1.2 (http://java.sun.com/j2ee/javaserverfaces/)	JSR 252 (http://jcp.org/en/jsr/detail?id=252)

TABLE 2–5 Major API changes on the Java EE 5 Platform *(Continued)*

API	JSR
JavaServer Pages 2.1 (http://java.sun.com/products/jsp/)	JSR 245 (http://jcp.org/en/jsr/detail?id=245)
JavaServer Pages Standard Tag Library 1.2 (http://java.sun.com/products/jsp/jstl/)	JSR 52 (http://jcp.org/en/jsr/detail?id=52)
Management Technologies	
J2EE Management (http://java.sun.com/j2ee/tools/management/)	JSR 77 (http://jcp.org/en/jsr/detail?id=77)
J2EE Application Deployment (http://java.sun.com/j2ee/tools/deployment/)	JSR 88 (http://jcp.org/en/jsr/detail?id=88)
Java Authorization Contract for Containers (http://java.sun.com/j2ee/javaacc/)	JSR 115 (http://jcp.org/en/jsr/detail?id=115)
Other Java EE Technologies	
Common Annotations for the Java Platform	JSR 250 (http://jcp.org/en/jsr/detail?id=250)
Java Transaction API (JTA) (http://java.sun.com/products/jta/)	JSR 907 (http://jcp.org/en/jsr/detail?id=907)
JavaBeans Activation Framework (JAF) 1.1 (http://java.sun.com/products/javabeans/glasgow/jaf.html)	JSR 925 (http://jcp.org/en/jsr/detail?id=925)
JavaMail (http://java.sun.com/products/javamail/)	JSR 919 (http://jcp.org/en/jsr/detail?id=919)
Java Message Service API (http://java.sun.com/products/jms/)	JSR 914 (http://www.jcp.org/en/jsr/detail?id=914)
Java Persistence API (http://java.sun.com/j2ee/persistence/faq.html)	JSR 220 (http://www.jcp.org/en/jsr/detail?id=220)

Java EE 5 SDK

Sun Java System Application Server 9 is available as part of the Java EE 5 SDK.

There are three Java EE 5 SDK versions:

- Java EE 5 SDK (http://java.sun.com/javaee/downloads/)
- Java EE 5 SDK SOA Starter Kit Preview (http://java.sun.com/integration/openesb/releasenotes.jsp)
- Java EE 5 Tools Preview Bundle (http://www.netbeans.info/downloads/download.php?type=5.5)

In addition to Application Server, the Java EE 5 SDK includes Java 2 Platform Standard Edition 5.0 Update 6, Java EE 5 Samples, Java BluePrints, and API documentation (Javadoc).

Java EE 5 SDK SOA Starter Kit Preview also includes the Project Open ESB Starter Kit that implements an enterprise service bus (ESB) runtime based on the JSR 208 specification for Java Based Integration (JBI). It contains a BPEL service engine, Java EE service engine, HTTP SOAP binding component, and examples of how to use the BPEL service engine.

Java EE 5 Tools Preview Bundle adds NetBeans 5.5 Beta, NetBeans Enterprise Pack 5.5, and Sun Java System Access Manager 7.1, but not J2SE 5.0.

For more information on the other SDK components, see:

- Project Open ESB Starter Kit
- NetBeans 5.5
- NetBeans Enterprise Pack 5.5

Switching to Another Supported Java Version

Sun Java System Application Server 9 requires J2SE 5.0 or greater as the underlying JVM. If you want to switch from one Java version to another, perform the following general steps. (Windows and Unix)

▼ To switch to another supported Java version

1 **Download the Java SDK (not the JRE) and install it on your system, if you have not already done so.**

The Java SDK can be downloaded from `http://java.sun.com/j2se`.

2 **Completely stop the Application Server.**

You can use the following command line:

```
as-install/bin/asadmin stop-domain
```

Alternatively, you can use the Administration Console GUI:

a. **Click the Application Server node.**

b. **Click** *Stop Instance***.**

3 **Edit the** `install_dir/config/asenv.conf` **file (**`asenv.bat` **on Windows), changing the value for** `AS_JAVA` **to point to the new J2SE home directory.**

4 **Edit the** `as-install/samples/common.properties` **file, changing the line beginning** `com.sun.aas.javaRoot...` **to reference the new J2SE home directory.**

5 Restart the Application Server.

```
as-install/bin/asadmin start-domain
```

◆ ◆ ◆ CHAPTER 3

Known Issues and Limitations

This section describes known problems and associated workarounds for the Sun Java System Application Server Platform Edition 9 product. If a summary statement does not specify a particular platform, the problem applies to all platforms. This information is organized into the following sections:

Administration

The `package-appclient` script does not work if `domain1` is not present. (ID 6171458)

By default, there is a hard-coded value in `$INSTALL/lib/package-appclient.xml` for the `AS_ACC_CONFIG` variable for `domain1` that is pointed to by `asenv.conf`. If `domain1` is deleted and a new domain created, the `AS_ACC_CONFIG` variable is not updated with the new domain name, which causes the `package-appclient` script to fail.

Solution

Do one of the following:

- Leave `domain1` intact, and create your other domains around it.
- Remove `domain1` and replace the hard-coded value for `domain1` in `$INSTALL/lib/package-appclient.xml` with the new domain name. This will have to be done every time a new domain is created if `domain1` is not present.

Starting Application Server with additional JMX Agent is not supported. (ID 6200011)

J2SE 1.4.*x*, 5.0, or later can be configured on the Application Server. An integral feature of J2SE 5.0 platform is the ability to start a JMX agent. This is activated when you explicitly set system properties at the server startup.

Example values include:

```
name="com.sun.management.jmxremote" value="true"
name="com.sun.management.jmxremote.port" value="9999"
name="com.sun.management.jmxremote.authenticate" value="false"
name="com.sun.management.jmxremote.ssl" value="false"
```

After configuring JMX properties and starting the server, a new `jmx-connector` server is started within the Application Server VM. An undesirable side-effect of this is that the administration functions are affected adversely, and the Application Server administration GUI and CLI may produce unexpected results. The problem is that there are some conflicts between the built in `jmx-connector` server and the new `jmx-connector` server.

Solution

If using `jconsole` (or any other JMX-compliant client), consider reusing the standard JMX Connector Server that is started with Application Server startup.

When the server starts up, a line similar to the one shown below appears in the server.log. You can connect to the `JMXServiceURL` specified there and perform the same management/configuration operations after successfully providing the credentials; for example:

```
[#|2004-11-24T17:49:08.203-0800|INFO|sun-appserver-ee8.1|javax.enterprise.
system.tools.admin|_ThreadID=10;|ADM1501: Here is the JMXServiceURL for the
JMXConnectorServer: [service:jmx:rmi:///jndi/rmi://hostname:8686/management/
rmi-jmx-connector]. This is where the remote administrative clients should
connect using the JSR 160 JMX Connectors.|#]
```

For more information, refer to the *Sun Java System Application Server 9 Administration Guide*.

`display-error-statistics` brings negative stats

The Admin Console GUI is able to differentiate between a real zero (`0`) and a state for which statistics are not available (`N/A`). The command-line interface currently displays negative 1 (`-1`) instead of `N/A`. For example, the following output for the `display-error-statics` command shows –1 instead of `N/A`.

```
# asadmin display-error-statistics
Please enter the admin user name>admin
Please enter the admin password>
Timestamp                              Severity  Warning
```

```
------------------------------------------------------------
1143659837750(Mar 29, 2006 11:17:17 AM)     0        0
1143656237750(Mar 29, 2006 10:17:17 AM)     0        0
1143652637750(Mar 29, 2006 9:17:17 AM)      -1       -1
1143649037750(Mar 29, 2006 8:17:17 AM)      -1       -1
1143645437750(Mar 29, 2006 7:17:17 AM)      -1       -1
------------------------------------------------------------
Command display-error-statistics executed successfully.
```

Solution

Use the Admin Console GUI to display error statistics. This issue will be fixed in a later release.

Application Client

This section describes known application client issues and associated solutions.

Library JAR packaged in Application Client Archive overwrites MANIFEST file. (ID 6193556)

If you have a top level JAR file inside your client JAR (in this case, reporter.jar), when you deploy the client JAR, the MANIFEST file for that JAR overwrites the MANIFEST file for the client JAR.

Solution

None at this time.

Windows platform – `APPCPATH` is not being set in the classpath (ID 6419847)

It has been observed that the `appclient.bat` command is not accepting the `APPCPATH` when running on Windows. The problem can be traced to quotes around the `APPCPATH` statement in the `apppclient.bat` file. For example, the following statement fails:

```
set CLASSPATH="%JAXP_IMPL_JARS%";%JVM_CLASSPATH%
if not %APPCPATH%x == x set CLASSPATH=%CLASSPATH%;"%APPCPATH%"
```

Whereas the following statement works correctly:

```
set CLASSPATH=%JAXP_IMPL_JARS%;%JVM_CLASSPATH%
if not %APPCPATH%x == x set CLASSPATH=%CLASSPATH%;%APPCPATH%
```

Solution

Remove the double quotes that surround the references to `APPCPATH` (and `JAXP_IMPL_JARS`).

Installation

This section describes known installation/uninstallation issues and associated solutions.

Installation shutdown hanging on some Linux systems after clicking the Finish button. (5009728)

This problem has been observed on several Linux systems. It is most common on Java Desktop System 2 but has also been observed on RedHat distributions.

After clicking the Finish button on the last installer screen, the installer fails to launch a browser window containing the product About page or product registration page, and hangs indefinitely, not returning the command prompt.

Solution

Exit the installer by pressing Ctrl+C in the terminal window in which the installer was started. After doing this, browser window containing product About page or registration page will sometimes be launched, but if it does not show up, start the browser and enter following URL in order to review About page:

`file://`*`install_dir`*`/docs/about.html`

If you also selected the installation option to register the product, follow the link to registration page available on product About page.

Intermittent J2SE detection and bootstrap issues in install wrapper on Linux. (6172980)

The `setup` executable that launches the Linux installer sometimes hangs. Instead of resolving the J2SE location and starting the install wizard, the wrapper hangs and returns the following messages:

```
Chcking available disk space....
Checking Java(TM) 2 Runtime Environment....
Extracting Java(TM) 2 Runtime Environment....
Deleting temporary files.....
```

This issue is seen only in some versions of Linux, and seems to depend on environment settings, especially the presence of the `JAVA_HOME` variable.

Solutions

To work around this issue:

▼ To work around the bootstrap issues on Linux

1. **Unset the `JAVA_HOME` variable by running `unset` or `unsetenv` depending on your shell.**
2. **Run `setup` with the `-javahome` option to specify the `JAVA_HOME` used by the installer.**

Lifecycle Management

This section describes known lifecycle management issues and associated solutions.

After setting the `ejb-timer-service` property `minimum-delivery-interval` to `9000`, an attempt to set the `ejb-timer-service` property `redelivery-interval-in-mills` to `7000` causes the `set` command to fail with the following error: (ID 6193449)

```
[echo] Doing admin task set
[exec] [Attribute(id=redelivery-interval-internal-in-millis) : Redelivery-
Interval (7,000) should be greater than or equal to Minimum-delivery-
interval-in-millis (9,000)]
[exec] CLI137 Command set failed.
```

- `minimum-delivery-interval` is the minimal interval duration between deliveries of the same periodic timer.
- `redelivery-interval-in-mills` is the time the timer service will wait after a failed `ejbTimeout` before attempting redelivery.

  The problem is that the logic that relates the redelivery interval property to the minimum delivery property is incorrect and prevents you from using the GUI or the CLI to set any value where the minimum delivery interval is greater than redelivery interval.

  The `minimum-delivery-interval-in-millis` must always be set equal to or higher than ejb-timer-service property `redelivery-interval-in-millis`. The problem is that there is an erroneous validation check in the Application Server to verify that the value for `redelivery-interval-in-millis` is greater than the value for `minimum-delivery-interval-in-millis`.

Solution

Use the default values for these properties, as follows:

```
minimum-delivery-interval(default)=7000
redelivery-interval-in-millis(default)=5000
```

Values other than these defaults will generate an error.

Java Persistence API

This section lists known issues and associated solutions related to the Java Persistence API.

An `UPDATE` or `DELETE` query using a subquery in the `WHERE` clause results in a `NullPointerException` during query compilation.

See `https://glassfish.dev.java.net/issues/show_bug.cgi?id=572` for more information.

Solution

Do not use this type of query.

The query compiler does not check all the rules as defined in the Java Persistence Language specification.

The query compiler does not check all the rules as defined in the Java Persistence Language chapter of the specification. In particular, it does not check:

- Type compatibility of function arguments.
- Type compatibility of operands of arithmetic, logical, and comparison operators
- `SELECT` clause requirements for an `ORDER BY` query.

An invalid query may compile but may cause a `SQLException` at runtime. Or it might happen that the underlying database is less restrictive and supports the generated SQL, but executing the same query on a different database fails with a `SQLException`.

Solution

Verify the above conditions manually.

A query grouping by a `JOIN` variable and directly selecting the `JOIN` variable might result in a `SQLException`

A query grouping by a `JOIN` variable and directly selecting the `JOIN` variable might result in a `SQLException` complaining about non-grouping expressions being selected. For more information, see `https://glassfish.dev.java.net/issues/show_bug.cgi?id=197`. For example:

```
SELECT AVG(o.totalPrice), c FROM Order o JOIN o.customer c GROUP BY c
```

Solution

A possible workaround is to directly navigate the relationship instead of defining a `JOIN` variable; for example:

```
SELECT AVG(o.totalPrice), o.customer FROM Order GROUP BY o.customer
```

In some cases, the query compiler throws a misleading error message, because it mentions the wrong token in the error message.

For more information, see `https://glassfish.dev.java.net/issues/show_bug.cgi?id=550`.

Solution

Manually verify what is wrong in the query.

Two persistence units containing the same class cannot currently be deployed in the same EAR file.

Solution

Use different class names.

You cannot list `MappedSuperclass` explicitly using class element in `persistence.xml`.

For more information, see
https://glassfish.dev.java.net/issues/show_bug.cgi?id=558.

Solution

Do not list `MappedSuperclass` explicitly.

Entity mapping `@ManyToMany` fails on inherited class.

For more information, see
https://glassfish.dev.java.net/issues/show_bug.cgi?id=578.

Solution

Do not use `ManyToMany` relationship in a subclass.

Using `List<>` for a relationship causes `StringIndexOutOfBoundException` in `MetadataHelper.getAttributeNameFromMethodName`.

For more information, see
https://glassfish.dev.java.net/issues/show_bug.cgi?id=557.

Solution

Use `java.util.Collection` instead.

Unable to map null database values to primitives.

Solution

Use Java wrapper types for mapping to nullable database columns.

Accessing a `LAZY` initialized relationship from client side sometimes causes a `NullPointerException`.

For more information, see
https://glassfish.dev.java.net/issues/show_bug.cgi?id=404.

Solution

Access `LAZY` initialized relationship in server-side code before returning an instance to the client.

A query selecting a relationship field does not include null in the query result.

A query selecting a relationship field does not include null in the query result when the value of the relationship field is null. This entry is instead skipped from the query result; for example:

```
SELECT o.customer FROM Order o WHERE ...
```

See https://glassfish.dev.java.net/issues/show_bug.cgi?id=637 for more information.

Solution

Select a state field from the related instance.

```
SELECT o.customer.customerId FROM Order o WHERE ...
```

A query selecting a JOIN identification variable defined for a single valued relationship field may result in invalid SQL.

For example:

```
SELECT c FROM Order o LEFT OUTER JOIN o.customer c
```

See https://glassfish.dev.java.net/issues/show_bug.cgi?id=638 for more information.

Solution

Select a state field from the related instance.

```
SELECT c.customerId FROM Order o LEFT OUTER JOIN o.customer c
```

EntityManager.find() erroneously throws an IllegalArgumentException

`EntityManager.find()` erroneously throws an `IllegalArgumentException` for an entity that is a subclass of another entity if the primary key class is defined by an `@IdClass` annotation. See https://glassfish.dev.java.net/issues/show_bug.cgi?id=595 for more information.

Solution

Use the class of the topmost superclass as the argument to the `find()` method, and cast the result to the subclass.

Cannot persist an entity with relationship field set to a java.util.HashSet.

See https://glassfish.dev.java.net/issues/show_bug.cgi?id=643 for more information.

Solution

Use `java.util.ArrayList` as the initial value.

JavaServer Faces

This section describes known JavaServer Faces (JSF) issues and associated solutions. Note that you can find additional information about the JSF project and submit any JSF bugs you may encounter by going to the JSF project (https://javaserverfaces.dev.java.net/) Web site.

JSF 1.1 - 1.2 Spec Violation: Variable and Property Resolvers (ID 6419278)

JavaServer Faces Technology Applications that use the `VariableResolver` decoration to extend the functionality of the Expression Language may not work properly.

Section 10.4.5 of the JavaServer Faces Technology Specification (`http://jcp.org/en/jsr/detail?id=252`) states:

"When providing a replacement for the default `PropertyResolver`, `VariableResolver`, `ActionListener`, `NavigationHandler`, `ViewHandler`, or `StateManager`, the decorator design pattern is leveraged, so that if you provide a constructor that takes a single argument of the appropriate type, the custom implementation receives a reference to the implementation that was previously fulfilling the role. In this way, the custom implementation is able to override just a subset of the functionality (or provide only some additional functionality) and delegate the rest to the existing implementation."

In Application Server 9, a custom `VariableResolver` implementation will receive a "previous" `VariableResolver` that does not fully fulfill the role of variable resolution.

Solution

Instead of delegating to the "previous" `VariableResolver` to resolve an expression, we recommend creating an `ValueExpression` and evaluating it.

EXAMPLE 3–1 Evaluating a `ValueExpression`

```
public class CustomVR extends VariableResolver {

    private VariableResolver previous = null;

    public CustomVR(VariableResolver previous) {
    this.previous = previous;
    }

    public Object resolveVariable(FacesContext context, String name)
    throws EvaluationException {
    Object result = null;

    // Do some action that may resolve the variable.  If not, you
    // may be tempted to simply do:

    // result = previous.resolveVariable(context, name);

    // But this would not work due to bug 6419278.  A fix is
    // available, please see the Release Notes.  However, a
    // workaround is the following.

    ValueExpression ve = context.getApplication().getExpressionFactory(). \
```

EXAMPLE 3–1 Evaluating a `ValueExpression` *(Continued)*

```
createValueExpression(context.getELContext(), "#{" + name + "}", Object.class);
    try {
        result = ve.getValue(context.getELContext());
    }
    catch (PropertyNotFoundException pnfe) {
        throw new EvaluationException(pnfe);
    }
    catch (ELException ele) {
        throw new EvaluationException(ele);
    }
    return result;

    }
}
```

Upgrade

This section describes known issues and associated solutions related to upgrading and using the Upgrade Tool.

Domains created in custom-path other than *install_dir*`/domains` directory are not upgraded directly while upgrading from Application Server Platform Edition 8 to Application Server Platform Edition 9. (ID 6165528)

When running the Upgrade Utility and identifying the *install_dir* as the source installation directory, the upgrade process upgrades only those domains that are created under *install_dir*`/domains` directory. Domains created in other locations are not upgraded.

Solution

Before starting the upgrade process, copy all the domain directories from their different locations to the *install_dir*`/domains` directory.

The installer running "Upgrade in place" fails to start upgrade tool on some Linux systems after clicking on the "Start Upgrade Wizard" button. (6207337)

This problem has been observed on several Linux systems, it is most common on Java Desktop System 2 but has also been observed on RedHat distributions.

After clicking the Start Upgrade Tool button on the final installer screen, the installer fails to launch the upgrade tool to complete the upgrade process, and hangs indefinitely, not returning the command prompt.

Solution

This issue is not encountered if command line installation mode is used to run upgrade in place.

▼ To use command line installation mode

1 **If you ran upgrade in place in GUI mode and encountered this problem, exit the installer by pressing Ctrl+C in the terminal window in which the installer was started.**

2 **Start upgrade tool from the terminal window, using following command:**

```
install_dir/bin/asupgrade --source install_dir/domains --target install_dir
--adminuser adminuser--adminpassword adminpassword --masterpassword changeit
```

adminuser and *adminpassword* should match the values used for the installation you are upgrading.

3 **When the upgrade tool completes the upgrade process you can also start the browser and enter following URL in order to review About page:**

```
file://install_dir/docs/about.html
```

If you also selected the installation option to register the product, follow the link to registration page available on product About page.

Upgrade Tool does not transfer all JVM options (ID 6409650)

The Upgrade Tool does not transfer JVM options, such as those related to stack size and heap size because they should retain the values specified during installation. This is because the values for these options are likely to be set for specific reasons for the current version of the product. The Upgrade Tool will log a message indicating which options have not been transferred.

Copy static documents in `docroot` directory after upgrade (ID 6409641)

After you upgrade your Application Server installation, copy any static documents in the docroot directory to the new docroot directory. If you did an in-pace upgrade, copy the documents from the backed-up domain directory under the `domaindir` of the installation.

Must close Application Server files when running Upgrade Tool on Windows

While running Upgrade Tool on Windows, you must close all files in the installation being upgraded. If a file is left open, then the Upgrade Tool has been observed to hang.

Username Token not encrypted if we use `XWSClientProvider` and `XWSServerProvider`. (ID 6409652)

To use the XWSCLientProvider and XWSServerProvider in an upgraded Application Server , you must modify the deployment descriptors of the applications that use these providers, to refer to XWSClientProvider and XWSServerProvider.

The deployment decriptors that you must change are `sun-web.xml` and `sun-ejb-jar.xml`. The Upgrade Tool logs a message in the upgrade log if it finds ClientProvider and ServerProvider in the `sun-web.xml` or `sun-ejb-jar.xml`.

Web Container

This section describes known web container issues and associated solutions.

Unable to deploy WAR with Servlet 2.4-based `web.xml` that contains an empty `<load-on-startup/>` element. (ID 6172006)

The optional `load-on-startup` Servlet element in a `web.xml` indicates that the associated Servlet is to be loaded and initialized as part of the startup of the Web application that declares it.

Note – This limitation applies to Servlet 2.4-based `web.xml` only; it is possible to specify an empty `load-on-startup` element using a Servlet 2.5-based `web.xml`.

The optional content of this element is an integer indicating the order in which the Servlet is to be loaded and initialized with respect to the Web application's other Servlets. An empty `<load-on-startup/>` indicates that the order is irrelevant, as long as the Servlet is loaded and initialized during the startup of its containing web application.

The Servlet 2.4 schema for `web.xml` no longer supports an empty `<load-on-startup/>`, meaning that an integer must be specified when using a Servlet 2.4 based `web.xml`. If specifying an empty `<load-on-startup/>`, as in `<load-on-startup/>`, the `web.xml` will fail validation against the Servlet 2.4 schema for `web.xml`, causing deployment of the Web application to fail.

Backwards compatibility issue: Specifying an empty `<load-on-startup/>` still works with Servlet 2.3 based `web.xml`.

Solution

Specify `<load-on-startup/>0</load-on-startup/>` when using a Servlet 2.4 based `web.xml` to indicate that Servlet load order does not matter.

Unable to compile JSP page on resource constrained servers. (ID 6184122)

The JSP page is accessed but fails to compile, and the server log contains the error message "`Unable to execute command`" with the following stack trace:

```
at org.apache.tools.ant.taskdefs.Execute$Java13CommandLauncher.exec
(Execute.java:655) at org.apache.tools.ant.taskdefs.Execute.launch
(Execute.java:416) at org.apache.tools.ant.taskdefs.Execute.execute
(Execute.java:427) at org.apache.tools.ant.taskdefs.compilers.
DefaultCompilerAdapter.executeExternalCompile(DefaultCompilerAdapter.
java:448) at org.apache.tools.ant.taskdefs.compilers.JavacExternal.
execute(JavacExternal.java:81) at org.apache.tools.ant.taskdefs.Javac.
compile(Javac.java:842) at org.apache.tools.ant.taskdefs.Javac.execute
(Javac.java:682) at org.apache.jasper.compiler.Compiler.generateClass
(Compiler.java:396)
```

Solution

Set the JSP compilation switch `fork` to `false`.

This can be done either of two ways:

- Globally, by setting the `fork` `init` parameter of the `JspServlet` in `${S1AS_HOME}/domains/domain1/config/default-web.xml` to `false`:

  ```
  <servlet\> <servlet-name\>jsp</servlet-name\> <servlet-class\>org.apache.
  jasper.servlet.JspServlet</servlet-class\> .... <init-param\> <param-name\>
  fork</param-name\> <param-value\>false</param-value\> </init-param\> ....
  </servlet\>
  ```

- On a per-web application basis, by setting the `fork` JSP configuration property in `sun-web.xml` to `false`:

  ```
  <sun-web-app\> <jsp-config\> <property name="fork" value="false" /\>
  </jsp-config\> </sun-web-app\>
  ```

Either setting will prevent `ant` from spawning a new process for `javac` compilation.

ParserUtils picks up parser from web app (ID 6412405)

When accessing a JSP, the following exception may be logged under certain circumstances:

```
java.lang.AbstractMethodError: org.apache.xerces.dom.DeferredDocumentImpl.
setDocumentURI(Ljava/lang/String;)V
```

Solution

Add the following system property to your `domain.xml` file:

```
<jvm-options>-Djavax.xml.parsers.DocumentBuilderFactory=com.sun.org.apache.xerces. \
internal.jaxp.DocumentBuilderFactoryImpl</jvm-options>
```

Alternatively, remove the JAR file containing the `org.apache.xerces` package form your Web application's `WEB-INF/lib`.

TLDs in EAR-bundled JARs not found. (Glassfish Issue 590)

JAR-packed JSP tag libraries bundled in EAR files cannot be referenced by the JSP pages of a WAR file that is part of the EAR, even if the `META-INF/MANIFEST.MF` resources of that WAR file lists the tag library JAR file(s) in its `Class-Path` manifest attribute.

When accessing a JSP page that imports a tag library packaged in one the EAR file's JAR files, an error message similar to the following will appear in the server log and the JSP page will fail to compile:

```
The absolute uri: <taglib_uri> cannot be resolved in either web.xml or
the jar files deployed with this application
```

Solution

Bundle any JSP tag library JAR files in the `WEB-INF/lib` directory of any WAR file whose JSP pages import those tag libraries.

More information about this bug can be found on the Glassfish site at `https://glassfish.dev.java.net/issues/show_bug.cgi?id=590`.

Platform-specific Issues

Must use fully qualified domain names when setting denyRemoteHost and allowRemoteHost properties on Windows XP and Windows Server 2003 (ID 6377272)

On Windows systems, you must use fully qualified domain names (for example, `myhost.mydomain.com`) instead of the short name (for example, just `myhost`) for Windows hosts in setting IP-based request filtering properties, even though Windows commands such as `hostname` and `ipconfig /all` and Java the API `InetAddress.getLocalHost().getHostName()` return short host names.

For example, the following command denies access by Windows host `foobar`:

```
asadmin --user adminuser --password adminpasswd
--echo server.http-service.virtual-server.server.property.denyRemoteHost=foobar.sun.com
```

Note that the host name is fully-qualified with the DNS domain name, `foobar.sun.com`.