



Sun Java System Application Server 9.1 Release Notes



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

Part No: 819-3666-15
January 2010

Copyright 2010 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. All rights reserved.

Sun Microsystems, Inc. has intellectual property rights relating to technology embodied in the product that is described in this document. In particular, and without limitation, these intellectual property rights may include one or more U.S. patents or pending patent applications in the U.S. and in other countries.

U.S. Government Rights – Commercial software. Government users are subject to the Sun Microsystems, Inc. standard license agreement and applicable provisions of the FAR and its supplements.

This distribution may include materials developed by third parties.

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

Sun, Sun Microsystems, the Sun logo, the Solaris logo, the Java Coffee Cup logo, docs.sun.com, Java, Java, and Solaris are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

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

Products covered by and information contained in this publication are controlled by U.S. Export Control laws and may be subject to the export or import laws in other countries. Nuclear, missile, chemical or biological weapons or nuclear maritime end uses or end users, whether direct or indirect, are strictly prohibited. Export or reexport to countries subject to U.S. embargo or to entities identified on U.S. export exclusion lists, including, but not limited to, the denied persons and specially designated nationals lists is strictly prohibited.

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

Copyright 2010 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. Tous droits réservés.

Sun Microsystems, Inc. détient les droits de propriété intellectuelle relatifs à la technologie incorporée dans le produit qui est décrit dans ce document. En particulier, et ce sans limitation, ces droits de propriété intellectuelle peuvent inclure un ou plusieurs brevets américains ou des applications de brevet en attente aux Etats-Unis et dans d'autres pays.

Cette distribution peut comprendre des composants développés par des tierces personnes.

Certains composants de ce produit peuvent être dérivés du logiciel Berkeley BSD, licenciés par l'Université de Californie. UNIX est une marque déposée aux Etats-Unis et dans d'autres pays; elle est licenciée exclusivement par X/Open Company, Ltd.

Sun, Sun Microsystems, le logo Sun, le logo Solaris, le logo Java Coffee Cup, docs.sun.com, Java, Java et Solaris sont des marques de fabrique ou des marques déposées de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays. Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

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

Les produits qui font l'objet de cette publication et les informations qu'il contient sont régis par la législation américaine en matière de contrôle des exportations et peuvent être soumis au droit d'autres pays dans le domaine des exportations et importations. Les utilisations finales, ou utilisateurs finaux, pour des armes nucléaires, des missiles, des armes chimiques ou biologiques ou pour le nucléaire maritime, directement ou indirectement, sont strictement interdites. Les exportations ou réexportations vers des pays sous embargo des Etats-Unis, ou vers des entités figurant sur les listes d'exclusion d'exportation américaines, y compris, mais de manière non exclusive, la liste de personnes qui font objet d'un ordre de ne pas participer, d'une façon directe ou indirecte, aux exportations des produits ou des services qui sont régis par la législation américaine en matière de contrôle des exportations et la liste de ressortissants spécifiquement désignés, sont rigoureusement interdites.

LA DOCUMENTATION EST FOURNIE "EN L'ETAT" ET TOUTES AUTRES CONDITIONS, DECLARATIONS ET GARANTIES EXPRESSES OU TACITES SONT FORMELLEMENT EXCLUES, DANS LA MESURE AUTORISEE PAR LA LOI APPLICABLE, Y COMPRIS NOTAMMENT TOUTE GARANTIE IMPLICITE RELATIVE A LA QUALITE MARCHANDE, A L'APTITUDE A UNE UTILISATION PARTICULIERE OU A L'ABSENCE DE CONTREFACON.

Overview

The Sun Java™ System Application Server 9.1 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 9.1 is free for development, deployment and redistribution. Customers interested in redistribution should contact [Sun OEM](#) sales for a redistribution license.

The Sun Java System Application Server 9.1 product greatly simplifies the task of creating and administering Web services applications. It provides superior performance, clustering, and high availability features for scalable services that continue to operate despite software and hardware faults. The Application Server provides a development path for web services that simplifies the development process while providing uniquely flexible growth opportunities.

- [“About These Notes” on page 3](#)
- [“Release Notes Revision History” on page 4](#)
- [“Accessibility Features” on page 4](#)
- [“Related Documentation” on page 5](#)
- [“How to Report Problems and Provide Feedback” on page 5](#)
- [“Sun Welcomes Your Comments” on page 6](#)
- [“Additional Sun Resources” on page 6](#)

About These Notes

These Release Notes contain important information available at the time of release of Sun Java System Application Server 9.1. Enhancements, known problems, and other late-breaking issues are addressed here. Read this document before you begin using Application Server 9.1.

The most up-to-date version of these release notes can be found at the Sun Java System [documentation Web site \(http://docs.sun.com/app/docs/coll/1343.4?q=glassfish\)](http://docs.sun.com/app/docs/coll/1343.4?q=glassfish). 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 changes that have been made in these release notes after the initial release of the Sun Java System Application Server 9.1 product.

TABLE 1-1 Release Notes Revision History

Revision Date	Description
September 2007	FCS release of the Sun Java System Application Server 9.1 product.
June 2008	Added System Virtualization Support section. Changed Sun Java System Web Server support to 6.1, 7.0.
November 2008	Added text on Message Queue version limitations in the Hardware and Software Requirements section. Removed Solaris 8 and Linux 2.1 as from supported Web Servers section.
December 2008	Added CR 6618442 to Known Issues, Administration section.
August 2009	Added RFE 6808339 to Known Issues, Documentation section.
January 2010	Added text to clarify supported platforms for 32-bit and 64-bit.

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

TABLE 1–2 Books in the Application Server Documentation Set

Book Title	Description
<i>Release Notes</i>	Late-breaking information about the software and the documentation. Includes a comprehensive, table-based summary of the supported hardware, operating system, JDK, and JDBC/RDBMS.
<i>Quick Start Guide</i>	How to get started with the Application Server product.
<i>Installation Guide</i>	Installing the software and its components.
<i>Deployment Planning Guide</i>	Evaluating your system needs and enterprise to ensure that you deploy the Application Server in a manner that best suits your site. General issues and concerns that you must be aware of when deploying the server are also discussed.
<i>Developer's Guide</i>	Creating and implementing Java 2 Platform, Enterprise Edition (J2EE™ platform) applications intended to run on the Application Server that follow the open Java standards model for J2EE components and APIs. Includes general information about developer tools, security, assembly, deployment, debugging, and creating lifecycle modules.
<i>J2EE 1.4 Tutorial</i>	Using J2EE 1.4 platform technologies and APIs to develop J2EE applications.
<i>Administration Guide</i>	Configuring, managing, and deploying Application Server subsystems and components from the Administration Console.
<i>High Availability Administration Guide</i>	Post-installation configuration and administration instructions for the high-availability database.
<i>Administration Reference</i>	Editing the Application Server configuration file, <code>domain.xml</code> .
<i>Upgrade and Migration Guide</i>	Migrating your applications to the new Application Server programming model, specifically from Application Server 6.x and 7. This guide also describes differences between adjacent product releases and configuration options that can result in incompatibility with the product specifications.
<i>Performance Tuning Guide</i>	Tuning the Application Server to improve performance.
<i>Troubleshooting Guide</i>	Solving Application Server problems.
<i>Error Message Reference</i>	Solving Application Server error messages.
<i>Reference Manual</i>	Utility commands available with the Application Server; written in man page style. Includes the <code>asadmin</code> command line interface.

How to Report Problems and Provide Feedback

If you have problems with Sun Java System Application Server, contact Sun customer support using one of the following mechanisms:

- [Feedback Submittal form \(http://java.sun.com/docs/forms/J2EE14SubmittalForm.html\)](http://java.sun.com/docs/forms/J2EE14SubmittalForm.html) — A form for submitting feedback on the Application Server product
- [Glassfish mailing lists \(https://glassfish.dev.java.net/servlets/ProjectMailingListList\)](https://glassfish.dev.java.net/servlets/ProjectMailingListList) — A variety of Glassfish community mailing lists for various interests and feedback
- [Bug database on Java Developer Connection \(http://developer.java.sun.com/servlet/SessionServlet?url=/developer/bugParade/index.jshtml\)](http://developer.java.sun.com/servlet/SessionServlet?url=/developer/bugParade/index.jshtml) — To view bugs or to submit a bug, use the Java Developer Connection Bug Parade
- [Java Technology Forums \(http://forum.java.sun.com/\)](http://forum.java.sun.com/) — An interactive message board for sharing knowledge and questions about Java technologies and programming techniques; use the Java EE 5 SDK forum for discussions related to the Sun Java System Application Server 9.1 product
- [Sun Software Support services \(http://www.sun.com/service/sunone/software\)](http://www.sun.com/service/sunone/software) — Links to the Knowledge Base, Online Support Center, and Product Tracker, as well as to maintenance programs and support contact numbers
- The telephone dispatch number associated with your maintenance contract
So that we can best assist you in resolving problems, please have the following information available when you contact support:
- Description of the problem, including the situation where the problem occurs and its impact on your operation
- Machine type, operating system version, and product version, including any patches and other software that might be affecting the problem
- Detailed steps on the methods you have used to reproduce the problem
- Any error logs or core dumps

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 9.1 Release Notes*, and the part number is 819-3666.

Additional Sun Resources

Useful information can be found at the following locations:

- [Application Server product information \(http://www.sun.com/software/products/appsrvr/home_appsrvr.html\)](http://www.sun.com/software/products/appsrvr/home_appsrvr.html)
- [Sun Java System Documentation \(http://docs.sun.com/prod/java.sys\)](http://docs.sun.com/prod/java.sys)
- [Sun Java System Professional Services \(http://www.sun.com/service/sunps/sunone\)](http://www.sun.com/service/sunps/sunone)
- [Sun Java System Software Products and Service \(http://www.sun.com/software\)](http://www.sun.com/software)
- [Sun Java System Software Support Services \(http://www.sun.com/service/sunone/software\)](http://www.sun.com/service/sunone/software)
- [Sun Java System Support and Knowledge Base \(http://www.sun.com/service/support/software\)](http://www.sun.com/service/support/software)
- [Sun Support and Training Services \(http://training.sun.com\)](http://training.sun.com)
- [Sun Java System Consulting and Professional Services \(http://www.sun.com/service/sunps/sunone\)](http://www.sun.com/service/sunps/sunone)
- [Sun Java System Developer Information \(http://developers.sun.com\)](http://developers.sun.com)
- [Sun Developer Support Services \(http://www.sun.com/developers/support\)](http://www.sun.com/developers/support)
- [Sun Java System Software Training \(http://www.sun.com/software/training\)](http://www.sun.com/software/training)
- [Sun Software Data Sheets \(http://www.sun.com/software\)](http://www.sun.com/software)
- [Sun Microsystems product documentation \(http://docs.sun.com/\)](http://docs.sun.com/)

About Application Server 9.1

The Sun Java System Application Server 9.1 is a Java EE 5 platform-compatible server for the development and deployment of Java EE applications and Java technology-based web services in large-scale production environments.

This chapter includes:

- “What’s New in the 9.1 Release” on page 9
- “Hardware and Software Requirements” on page 11
- “Java EE 5 Platform APIs” on page 21
- “Java EE 5 SDK” on page 23
- “Switching to Another Supported Java Version” on page 23
- “Known Java EE 5 Compatibility Issues” on page 24
- “Using the Application Server Update Center” on page 24
- “More About WSIT Integration” on page 25

What’s New in the 9.1 Release

The Application Server 9.1 includes the following enhancements:

- **Java EE 5 platform support** — Sun Java System Application Server Enterprise Edition 9.1 implements the Java EE 5 specification to deliver one of the best application runtimes for next-generation enterprise applications and web services. Application Server 9.1 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.

- **Web Services Interoperability Technologies (WSIT) support** – Sun is working closely with Microsoft to ensure interoperability of Web services enterprise technologies such as message optimization, reliable messaging, and security. The initial release of WSIT is a product of this joint effort. WSIT is an implementation of a number of open web services specifications to support enterprise features. In addition to message optimization, reliable messaging, and security, WSIT includes a bootstrapping and configuration technology. Starting with the core XML support currently built into the Java platform, WSIT uses or extends existing features and adds new support for interoperable web services, including:
 - Bootstrapping and Configuration
 - Message Optimization Technology
 - Reliable Messaging Technology
 - Security Technology

See [“More About WSIT Integration” on page 25](#) later in this chapter for more information about WSIT integration in Application Server 9.1.

- **Java Business Integration (JBI) support** – JBI extends Java EE with business integration Service Provider Interfaces (SPI). These SPI enable developers to create or implement a Java business integration environment for specifications such as WSCI, BPEL4WS and the W3C Choreography Working Group. A JBI implementation is installed directly by the Application Server installer, mostly in the `AS_HOME/jbi` directory. This directory contains all common JAR files and system components for the JBI including a `lifecycle` module that starts the JBI framework in the Application Server JVM.
- **In-memory replication support** – In-memory replication on other servers provides lightweight storage of session state data without the need to obtain a separate database, such as HADB. This type of replication uses memory on other servers for high availability storage of HTTP session and stateful session bean data. Clustered server instances replicate session state in a ring topology. Each backup instance stores the replicated data in memory. Replication of session state data in memory on other servers enables sessions to be distributed. The use of in-memory replication requires the GroupManagement Service (GMS) to be enabled (which is true by default).
- **Usage profiles** – Every administrative domain is associated with a usage profile, which identifies the capabilities of that domain. Application Server 9.1 provides the following profiles:
 - *Developer* – Use this profile if you are running your domain in a development environment and if your applications do not need clustering features, such as load balancing, high availability, and session replication. Note that the actual name of the profile is “`developer`” (case sensitive).

- *Cluster* – Use this profile if you want to create clusters of application server instances imparting scalability and high availability to the deployed Java EE applications. The state of the applications is persisted *in-memory*. Note that the actual name of the profile is “cluster” (case sensitive).
- *Enterprise* – Use this profile if you need HADB and NSS. This profile is not usable unless you install HADB and NSS separately or install the Application Server as part of the Java Enterprise System (Java ES).
- **Load balancing enhancements** – Several enhancements have been added to the load balancing plugin. Briefly, these include
 - *Weighted Round Robin* – An optional attribute called `weight` has been added to the instance `Loadbalancer.xml` file. This option enables the load balancer plugin to route requests according to the weight. For example, for every 500 requests, 100 will go to `instance1` and 400 would go to `instance2`. The default weight is 100. The weight is assigned to each instance from the admin console or command line, and the server `domain.xml` has an attribute for every instance indicating the weight.
 - *User-Defined Load Balancer Decision* – Enables users to define custom logic for load balancing; for example, user identity-based redirects and mime-based load balancing. This feature is implemented by means of a user-defined shared library that gets loaded by the load balancer. This custom shared library implements the interface as defined in `loadbalancer.h`, which is placed in `appserver_install_dir/lib/install/templates/`.
 - *Administration Enhancements* – Previous versions of Application Server required that the `loadbalancer.xml` file be manually copied to the particular server's config directory. Application Server 9.1 includes push automations for performing such copies between the Web server and Application Server. The load balancer itself is configured in the server's `domain.xml` file.
- **Open Source and GlassFish Community** — In June 2005, Sun launched the [GlassFish](http://java.sun.com/javaee/glassfish/) (<http://java.sun.com/javaee/glassfish/>) community with the goal of developing a free, open source, commercial-grade application server that implements the newest features of the Java EE 5 platform and related enterprise technologies. Application Server Enterprise Edition 9.1 is based on the source code developed by Sun engineers and the GlassFish community.
- **Update Center Support** – The Application Server 9.1 Update Center provides automated Application Server updates and easy access to additional components. See “[Using the Application Server Update Center](#)” on page 24 for more information.

Hardware and Software Requirements

This section lists the requirements that must be met before installing the Sun Java System Application Server 9.1 product.

- “[Supported Platforms](#)” on page 12
- “[System Virtualization Support](#)” on page 13

- “Important Patch Information” on page 13
- “JDK Version” on page 13
- “JDBC Drivers and Databases” on page 14
- “Using the Bundled Java DB Database” on page 15
- “Message Queue Versions” on page 18
- “Web Servers” on page 18
- “Browsers” on page 19
- “HADB Requirements and Supported Platforms” on page 19
- “Upgrading the Sun Java System Application Server” on page 20
- “Other Requirements” on page 20

Supported Platforms

All supported operating systems are 32-bit unless indicated otherwise. The 64-bit JDK is supported only on 64-bit supported operating systems.

The following table lists the operating systems with which the Sun Java System Application Server Enterprise Edition 9.1 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)	512 MB	512 MB	250 MB free	500 MB free	J2SE 5.0
Solaris 9, 10 (x86)					Java SE 6
64-bit Sun Solaris 10 (SPARC, x86)	512 MB	512 MB	250 MB free	500 MB free	J2SE 5.0
					Java SE 6
Redhat Enterprise Linux 3.0 U1, 4.0	512 MB	1 GB	250 MB free	500 MB free	J2SE 5.0
					Java SE 6
SuSE Linux Enterprise Server 10 SP1	512 MB	1 GB	250 MB free	500 MB free	J2SE 5.0
					Java SE 6

TABLE 2-1 Supported Operating Systems (Continued)

Operating System	Minimum Memory	Recommended Memory	Minimum Disk Space	Recommended Disk Space	JVM
Windows Server 2000 SP4+	1 GB	2 GB	500 MB free	1 GB free	J2SE 5.0
Windows 2000 Advanced Server SP4+					Java SE 6
Windows Server 2003					
Windows XP Pro SP1+					
Windows Vista					

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

Note – Its recommended that you 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

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.

JDK Version

The minimum (and certified) version of JDK required for Application Server 9.1 is **1.5.0_13**. JDK 1.5.0_13 can be downloaded from <http://java.sun.com/javaee/downloads/?intcmp=1282>.

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 Enterprise 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
Derby Network Client	Type 4	Derby 10.2
DataDirect 3.60.29	Type 4	Oracle 10g
DataDirect 3.6	Type 4	DB2 9.1 DB2 8.2
Oracle Thin	Type 4	Oracle 10g
PostGres	Type 4	8.1
MySQL Connector/J Driver 3.1	Type 4	MySQL 5

Table 2–3 lists all the additional databases and drivers scheduled to be supported in the Application Server FCS release.

TABLE 2–3 Supported JDBC Drivers and Databases (Application Server FCS)

JDBC Driver Vendor	JDBC Driver Type	Supported Database Server
Oracle OCI	Type 2	Oracle 9i
IBM DB2	Type 2	DB2 9.1
Oracle 10G	Type 4	-RAC
DataDirect 3.60.29	Type 4	Sybase ASE 15 MS SQL 2005
MySQL Connector/J Driver 3.1	Type 4	MySQL 5

In general, the Application Server Enterprise Edition 9.1 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 9.1. Java DB is based on the [Apache Derby database](#).

- “Starting and Stopping the Java DB Database” on page 15
- “Java DB Utility Scripts” on page 15
- “Exporting Tables from Pointbase to Java DB” on page 16

Starting and Stopping the Java DB Database

Sun Java System Application Server 9.1 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

Note – These scripts are deprecated in Application Server 9.1.

The Java DB configuration that ships with Application Server 9.1 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/) (<http://db.apache.org/derby/docs/10.1/tools/>) and [Admin](http://db.apache.org/derby/docs/10.1/adminguide/) (<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 sampled.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 \(http://db.apache.org/derby/docs/10.1/ref/\)](http://db.apache.org/derby/docs/10.1/ref/). Support for the Boolean datatype should be in the next release of Java DB.

Message Queue Versions

The embedded Sun Java System Message Queue code that is supplied as part of Application Server is only tested and certified (typically) against the equivalent version of the Message Queue broker. This means that using the supplied embedded Message Queue code to connect to a remote (not managed by Application Server) Message Queue broker running a different version of the Message Queue code is not supported.

Web Servers

This section lists the web servers that are supported for the Sun Java System Application Server 9.1.

TABLE 2-4 Supported Web Servers

Web Server	Version	Operating System
Sun Java System Web Server	6.1, 7.0	Solaris SPARC 9, 10
		Solaris x86 9, 10
		Red Hat Enterprise Linux , 3.0 Update 1

TABLE 2-4 Supported Web Servers (Continued)

Web Server	Version	Operating System
Apache Web Server	1.3+, 1.4, 2.0	Solaris SPARC 9, 10 Solaris x86 10 Red Hat Enterprise Linux 2.1 Update 2, 3.0 Update 1
Microsoft IIS™	5.0+	Windows Server 2003 Windows 2000 Advanced Server Service Pack 4+

Support for additional platforms, including Windows and HP-UX, will be available at a later date.

Browsers

This section lists the browsers that are supported with the Sun Java System Application Server 9.1.

TABLE 2-5 Supported Web Browsers

Browser	Version
Mozilla	1.7.12
Internet Explorer	6.0 Service Pack 2
Firefox	1.0.7
Safari	2
Netscape	8.0.4

HADB Requirements and Supported Platforms

In addition to the requirements listed in [“Hardware and Software Requirements” on page 11](#), verify that your system meets the requirements listed below for running HADB.

- [“Supported Platforms” on page 20](#)
- [“HADB Server Host Requirements” on page 20](#)
- [“HADB Management Host Requirements” on page 20](#)
- [“HADB Client Host Requirements” on page 20](#)

Note – HADB is only bundled with the Application Server 9.1 Enterprise profile. You can download HADB at ***. Also note that the Java components of the system have been built with JDK 5 and has been tested on JDK 1.5.

Supported Platforms

- **Solaris (SPARC)** – Solaris 8 MU7, Solaris 9 MU7, Solaris 10 RR.
- **Solaris (x86)** – Solaris 9 MU7, Solaris 10 RR.
- **RedHat Enterprise Linux** - 2.1 U5 (only ext2 file system is supported, not ext3), 3.0 U4 (both ext2 and ext3 are supported. Updates before U4 are not recommended due to excessive swapping). Note that HADB is tested on these operating system versions in 32-bit mode only. Also, note that HADB does not support RedHat Enterprise Linux 3.0 running in 64-bit mode due to a bug in the operating system (see known bug 6249685 in the [“High Availability” on page 37](#) section for details about impact on HADB).
- **Microsoft Windows** – Microsoft Windows 2000 Advanced Server Service Pack 4 and Microsoft Windows 2003 Enterprise Edition. Note that HADB does not support any of the forthcoming Microsoft Windows operating system versions in 64-bit mode.

HADB Server Host Requirements

- **Minimum memory** - 320 MB per node.
- **Minimum free disk space** - 70 MB for HADB binaries per host. In addition, disk space is needed for the data devices, 512 MB for a test installation per node.
- **Recommended memory** - 512 MB per node.
- **Recommended free disk space** - 70 MB for HADB binaries per host. In addition, disk space is needed for the data devices, 1200 MB for a test installation per node.

Note – Make sure write caching is disabled on devices storing HADB data and log files. Write caching is enabled by default on some Solaris platforms; for example, Solaris x86.

HADB Management Host Requirements

- **Minimum memory** - 128 MB
- **Minimum free disk space** - 70 MB for HADB binaries per node

HADB Client Host Requirements

- **Minimum memory** - 120 MB
- **Minimum free disk space** - 20 MB

Upgrading the Sun Java System Application Server

Refer to the [Sun Java System Application Server 9.1 Upgrade and Migration Guide](#) for complete instructions for upgrading from a previous version of the Application Server to the current version.

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 domain. By default, the initial default ports are:
 - 8080 for HTTP, 8181 for HTTPS
 - 3700 for IIOP, 3820 for IIOP/SSL and 3920 for IIOP/SSL with Mutual Authentication
 - 4848 (HTTP) for the Administration Server
 - 8686 (RMI) port for pure JMX Clients

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 9.1 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)** — Refer to the *Sun Java System Application Server 9.1 Upgrade and Migration Guide* for complete instructions for upgrading from a previous version of the Application 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 9.1 Upgrade and Migration Guide*.

Java EE 5 Platform APIs

The Sun Java System Application Server Enterprise Edition 9.1 supports the Java EE 5 platform. The following table lists the enhanced APIs available on the Java EE 5 platform.

TABLE 2-6 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/docs/api/)	JSR 244 (http://jcp.org/aboutJava/communityprocess/pr/jsr244/)

TABLE 2-6 Major API changes on the Java EE 5 Platform *(Continued)*

API	JSR
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 (https://jax-ws.dev.java.net/)	JSR 224 (http://jcp.org/en/jsr/detail?id=224)
Java API for XML-Based RPC (JAX-RPC) 1.1 (https://jax-rpc.dev.java.net/)	JSR 101 (http://jcp.org/en/jsr/detail?id=101)
Java Architecture for XML Binding (JAXB) 2.0 (https://jaxb.dev.java.net/)	JSR 222 (http://jcp.org/en/jsr/detail?id=222)
SOAP with Attachments API for Java (SAAJ) (https://saa.j.dev.java.net/)	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/javaxserverfaces/)	JSR 252 (http://jcp.org/en/jsr/detail?id=252)
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)

TABLE 2-6 Major API changes on the Java EE 5 Platform (Continued)

API	JSR
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 9.1 is available as part of the Java EE 5 SDK.

There are two Java EE 5 SDK versions:

- Java Application Platform SDK (<http://java.sun.com/javaee/downloads/>)
- Java EE 5 SDK (<http://java.sun.com/javaee/downloads/index.jsp>)

In addition, you can download these SDK distributions with the JDK or with the NetBeans 5.5.1 tools. For more information, access the download page at <http://java.sun.com/javaee/downloads/index.jsp>.

Note – GlassFish V2 and Application Server 9.1 *does not work* with NetBeans 5.5 for developing Java EE applications. You *must* use NetBeans 5.5.1.

Switching to Another Supported Java Version

Sun Java System Application Server 9.1 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)

Note – Downgrading to an earlier Java version is not recommended after a domain has been created with a newer Java VM. If you must downgrade your JVM, it is recommended that you do it on a per-domain basis. The following procedure describes how to do this.

▼ To switch to another supported Java version

- 1 **Download the desired 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 **Start the domain for which you want to change the JDK:**

```
as_install/bin/asadmin start-domain domain_name
```

- 3 **Log in to the Admin console and change the JVM attributes for the domain.**

In particular, you may want to change the JAVA_HOME variable on the JVM Settings page for the domain.

Alternatively, you can use the asadmin command:

```
as_install/bin/asadmin set "server.java-config.java-home=Path-To-Java-Home"
```

Known Java ES 5 Compatibility Issues

There are two known compatibility issues between Application Server 9.1 and Java ES 5 (JES5).

1. The JES5 Service Registry is not compatible with Application Server 9.1 because of the JSF 1.2 version provided by Application Server 9.1. The Service Registry needs to be upgraded to JES5u1 prior the Application Server upgrade to 9.1. This limitation is also documented in the Service Registry documentation.
2. The JES5 Portal Server is not compatible with Application Server 9.1 because of the JSF 1.2 version provided by Application Server 9.1. Portal Server needs to be upgraded to JES5u1 prior the 9.1 upgrade to 9.1.

On a Java ES 5 setup that has Portal Server on Application Server 8.2, upgrading Application Server 8.2 to 9.1 makes the Portal Server unusable. Application Server 9.1 uses JSF 1.2, but the JSF-Portlet bridge in Release 5 Portal Server does not support JSF 1.2. On Solaris/Linux the Portal Server needs to be upgraded to JavaES5 Update 1. On Windows do not upgrade Application Server to 9.1 if you want to continue using the Release 5 Portal Server, as Portal Server is not supported in Java ES 5 Update 1 on Windows.

Using the Application Server Update Center

The Application Server 9.1 Update Center provides automated Application Server updates and easy access to additional components.

When the Update Center is enabled, it performs an automated software update. During this automated update process, the Update Center collects and transmits the following data to Sun Microsystems (or its service provider):

- Unique installation ID (GUID)
- IP address
- Operating system information (name, version, architecture, locale)
- JDK version
- Module download information (module name, date, time, status, download time, number of bytes downloaded)

No personally identifiable information is tracked. No personally identifiable information is associated with any other data or used for reporting purposes.

▼ To Use the Update Center

To ensure explicit agreement of the automated update, the Update Center is disabled by default. To enable the Update Center to perform periodic checks and automated updates:

1 Start the Update Center.

- On Unix: `<installdir>/updatecenter/bin/updatetool`
- On Windows: `<installdir>\updatecenter\bin\updatetool.bat`

2 Select the Preferences tab.

3 In the Update Scheduling window, change the Check for Updates drop-down box value from Never (Manual) to a desired value. For example, daily or weekly.

4 Specify the desired day of the week and time of the day for the update.

5 Select the Save button to save your changes.

The Update Center will now automatically check for Glassfish or Application Server component updates according to the schedule specified. When an update is available, the Update Center will launch and notify you of the component available to update.

More About WSIT Integration

For detailed information about WSIT status, refer to the [WSIT Status Notes \(https://wsit.dev.java.net/source/browse/*checkout*/wsit/wsit/status-notes-1-0-FCS.html\)](https://wsit.dev.java.net/source/browse/*checkout*/wsit/wsit/status-notes-1-0-FCS.html) page. Also refer to the *The WSIT Tutorial* for information about using WSIT with Application Server.

Known Issues and Limitations

This chapter describes known problems and associated workarounds for the Sun Java System Application Server 9.1 software. 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” on page 27
- “Apache and Load Balancer Plugin” on page 33
- “Application Client” on page 34
- “Bundled Sun JDBC Drivers” on page 35
- “Documentation” on page 36
- “High Availability” on page 37
- “Installation” on page 46
- “Java EE Tutorial” on page 49
- “Java Persistence” on page 50
- “Lifecycle Management” on page 50
- “Logging” on page 52
- “Message Queue” on page 52
- “Monitoring” on page 53
- “Samples” on page 54
- “Security” on page 56
- “Upgrade Utility” on page 57
- “Web Container” on page 63
- “Web Server” on page 66
- “Web Services” on page 67

Administration

This section describes known administration issues and associated solutions.

The package-appclient script does not work if domain1 is not present (6171458)**Description**

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 `byasenv.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:

1. Leave `domain1` intact, and create your other domains around it.
2. 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.

Cannot restore backed-up domain with another name (6196993)**Description**

Mirroring of a domain on the same Application Server installation cannot be performed using the `backup-domain` and `restore-domain` commands because the domain cannot be restored using a different name than the original, even though the `asadmin restore-domain` command provides an option to rename the domain. Renaming the backed-up domain appears to succeed, but attempts to start the renamed domain fail because the entries in the domain configuration are not changed, and `startserv` and `stopserv` use the original domain name to set paths.

Solution

The domain name used for `restore-domain` must be the same as that used for the original `backup-domain` command. The `backup-domain` and `restore-domain` commands in Application Server 8.1 work only for backing up and restoring the same domain on the same machine.

Starting Application Server with additional JMX Agent is not supported (6200011)**Description**

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 Virtual Machine. An undesirable side-effect of this is that the administration functions are affected adversely, and the Application Server administration Console and command—line interface 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 JMXService URL 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 9.1 Administration Guide](#).

On UNIX, overly restrictive execute permissions on Application Server start and stop scripts (6206176)

Description

If you run the `asadmin restore-domain` command while logged in as user "A", the scripts will end up with permissions as 744 (`rwxr - - r - -`). If you subsequently attempt to start or stop a domain as user "B" (even if "B" is root), it will fail because the scripts are only executable for "A".

Solution

Change the permissions on the scripts:

```
chmod 755 appserv/domains/domain-name/bin/*
```

Load balancer configuration file does not get created with the endpoint URL of any web service (6236544, 6275436)

Description

When setting up the load balancer configuration with an application that has an EJB module that exports a web service URL, the context root for the web service isn't in the resulting `loadbalancer.xml` file.

Solution

1. Edit the `loadbalancer.xml` file to add the missing web module as follows:

```
<web-module context-root="context-root-name"
disable-timeout-in-minutes="30" enabled="true"/>
```

2. Replace `context-root-name` value with the context root name of the web service that was exposed as an EJB.

Application Server restart using `sun-appserv-admin` causes `LoginException` error (6288893)

Solution

1. Rename the existing `<as_install>/bin/asant` script to `asant.bak`.
2. Copy the `asant.template` file in `<as_install>/lib/install/templates/ee` (for SE/EE version) to the `<as_install>/bin/` directory and rename the file `asant`.
3. Edit the newly copied `<as_install>/bin/asant` script, replacing the `%CONFIG_HOME%` token with `<as_install>/config`.
4. If there were any manual changes made to the original `asant.bak` file, merge them into the new `asant` script.

`.asadmintruststore` file not described in the Application Server documentation (6315957)

Description

The `.asadmintruststore` file is not described in the Application Server documentation. If this file does not exist in the server administrator's home directory, you may experience serious bugs when upgrading certain applications hosted on the server.

Solution

- If possible, the `asadmin start-domain domain1` command should be run by user who installed the server.
- If it is not run by that user, the `.asadmintruststore` should be moved or copied from the home directory of installing user to the home directory of the running user.

- Note that if the file is moved (not copied) from the installing user's home directory to the running user's home directory, you might experience application upgrade problems, as described in bugs 6309079, 6310428 and 6312869, because the upgrade/install user (normally root in Java ES) will no longer have the `.asadmintruststore` file in his or her home directory.

Clustered instances fail to start due to a timeout in reaching the JMS broker (6523663)

Description

The default MQ integration mode for a Application Server cluster instance is LOCAL. When Application Server is installed in a location (PATH) that is long (read “not short”), `imqbrokersvc.exe` crashes when the cluster instance starts. The problem is a memory allocation problem in `imqbrokersvc`.

Solution

The JMS service type for the cluster instance must be changed from the default LOCAL to REMOTE. In this configuration, all the instances point back to the DAS broker. Follow the instructions below to configure a cluster in REMOTE mode.

Note – When using REMOTE mode, all instances are using one broker (DAS) , and therefore no broker cluster is created when the Application Server cluster starts up. See “Auto-clustering” in Section 4.1, Division iii of the one-pager at <http://www.glassfishwiki.org/gfwiki/attach/OnePagersOrFunctionalSpecs/as-mq-integration-gfv2.txt> for more information. The above functionality will not be available!

▼ Using the command-line

Before You Begin

Modify the port and password file according to your environment. Note that in the instructions below, the cluster name is `raccluster`, the DAS admin port is 5858, and the DAS JMS port is 7676.

1 Modify the cluster configuration, changing the JMS type to REMOTE.

```
$AS91_HOME/bin/asadmin.bat set --port 5858 --user admin --passwordfile \
$AS91_HOME/bin/password_file raccluster.jms-service.type=REMOTE
```

2 Create a JMS host corresponding to the DAS JMS host.

```
$AS91_HOME/bin/asadmin.bat create-jms-host --port 5858 --user admin --passwordfile \
$AS91_HOME/bin/password_file --target raccluster --mqhost localhost --mqport 7676 \
--mquser admin --mqpassword admin dashost
```

3 Set the default JMS host to be the DAS JMS host created in the previous step.

```
$AS91_HOME/bin/asadmin.bat set --port 5858 --user admin --passwordfile \
$AS91_HOME/bin/password_file raccluster.jms-service.default-jms-host=dashost
```

▼ Using the Admin GUI

- 1 Go to **Configurations->cluster_name-config->Java Message Service->JMS Hosts**.
- 2 Click **New** to create a new JMS host; name it **dashost**.
- 3 Enter configuration settings corresponding to the JMS service for the DAS; defaults are as follows:
 - Hostname: localhost
 - Port: 7676
 - Admin user: admin
 - Password: adminModify these settings as appropriate for your DAS JMS service.
- 4 Navigate back to the **Java Message Service** tab, and change the **JMS service type** to **REMOTE** (default is **LOCAL**).
- 5 Choose **dashost** from the **default-jms-host** drop-down list.
- 6 Save the changes, and then start your node-agent or cluster.

Cannot display jmaki chart in Netscape 8.1.3, Mozilla 1.7 and Safari 2.0.4 browsers (6543014)

Description

When trying to display a chart from the Log Statistics Monitoring page using some unsupported browsers, the following error may be thrown:

```
Error loading jmaki.widgets.jmaki.charting.line.Widget : id=form1:jmaki_chart11
Script: http://easqelx5.red.ipplanet.com:4848/resources/jmaki/charting/ \
line/component.js (line:5437).
Message: area.initialize is not a function
```

Solution

Use a supported browser. Refer to [“Browsers” on page 19](#) for a list of browsers supported by Application Server 9.1.

Default ports changing in each AS major release (6566481)

Description

The default admin port has changed in each of the past three major Application Server releases. Specifically, the default admin ports in 7.x, 8.x, and 9.x are as follows:

- AS 7.x: 4848

- AS 8.x: 4849
- AS 9.x: 4848

Solution

This is not a bug, but something to be aware of. The default admin port is just a recommendation. It is anticipated that future Application Server releases going forward will retain the default 4848 port.

On Windows: Application Server fails to start due to NSS initialization failure. (6618442)

Description

In some cases, Application Server (enterprise profile) might not start on Windows when PKCS11 fails to initialize.

Solution

Remove `softoken3.dll` from the Windows system path.

Apache and Load Balancer Plugin

This section describes known Apache Web server and load balancer plugin issues and associated solutions.

The High-Availability Administration Guide contains incorrect instructions for using `openssl` with Apache (6306784)

When compiling and building `openssl`, run the following commands:

```
cd openssl-0.9.7e
```

```
config
```

```
make
```

Also, for Apache 1.3, the directory name of the `mod_ssl` source will vary depending upon the release of Apache used. For example, for Apache 1.3.33, the name is `mod_ssl-2.8.22-1.3.33`.

The High-Availability Administration Guide does not contain instructions for using a certificate for Apache 2.0 (6307976)

To run Apache security, you must use a certificate. For instructions on obtaining a certificate from a certificate authority, see the information on certificates in the [modssl FAQ](http://www.modssl.org/docs/2.8/ssl_faq.html#ToC24) (http://www.modssl.org/docs/2.8/ssl_faq.html#ToC24).

Must start Apache Web Server as root (6308021)

On Solaris, if your Application Server was installed under root, you must start the Apache Web Server as root. Java Enterprise System installations are installed as root. For Apache 2.0, after starting as root, Apache switches and runs as another user you designate. You designate that user in the `/conf/httpd.conf` file. To start as root, on many systems you must edit the `httpd.conf` file to designate the correct group. Replace the line:

Group #-1

with

Group nobody

More information on user/group use is included in the `httpd.conf` file.

Application Client

This section describes known application client issues and associated solutions.

Library JAR packaged in Application Client Archive overwrites MANIFEST file (6193556)**Description**

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.

ACC always tries to connect to localhost:3700 (6527987)**Description**

The application client always tries to connect to `localhost:3700`. The problem is that several system properties need to be read before the client code is invoked.

Solution

Set the following as system properties (`-D` in your `JAVA_CMD`). Do *not* set them in your appclient code:

```
org.omg.CORBA.ORBInitialHost = server_instance_host  
org.omg.CORBA.ORBInitialPort = server_instance_port
```

Unable to start domain , missing sunpkcs11.jar (6571044)**Description**

Running on 64-bit Linux, the following exception when starting a domain. The issue is a missing sunpkcs11.jar under jdk1.5.0_11/jre/lib/ext/.

Solution

This is a known JDK bug with 64-bit Linux, and will be fixed in JDK 1.5.0_13.

ASQuickStartup **breaks** SocketChannel.keyFor(), **returning null instead of the** SelectionKey (Issue Tracker 3027)

Description

When a SocketChannel is registered on several Selectors, doing socketChannel.keyFor(lastRegisteredSelector) returns null instead of the SelectionKey.

Solution

This is related to a JDK bug, 6562829, and is expected to be fixed in 6.0 U3. A workaround has been included in Application Server 9.1, such that the selector is unwrapped before the keyFor API is called. This enables the keyFor to succeed until JDK bug is fixed.

Bundled Sun JDBC Drivers

This section describes known bundled Sun JDBC driver issues and associated solutions.

PreparedStatement errors (6170432)**Description 1**

If an application generates more than 3000 PreparedStatement objects in one transaction, the following error may occur with DB2:

```
[sunm][DB2 JDBC Driver] No more available statements.Please recreate your package with a larger dynamicSections value.
```

Solution 1

Add following properties to the connection pool definition to get the driver to rebind DB2 packages with a larger dynamic sections value:

```
createDefaultPackage=true replacePackage=true dynamicSections=1000
```

See the [Sun Java System Application Server 9.1 Administration Guide](#) for details about configuring connection pools.

Description 2

Related to the PreparedStatement error above, another error message that may be thrown is:

```
[sunm][DB2 JDBC Driver][DB2]Virtual storage or database resource is not available.
```

Solution 2

Increase the DB2 server configuration parameter *APPLHEAPSZ*. A good value is 4096.

Description 3

Isolation level TRANSACTION_SERIALIZABLE. If your application uses isolation level TRANSACTION_SERIALIZABLE and uses one of the parameters suggested above, it might hang while obtaining a connection.

Solution 3

To set desired isolation level for a connection, the corresponding connection pool has to be created at that isolation level. See the [Sun Java System Application Server 9.1 Administration Guide](#) for instructions.

Java DB is not started after machine reboot or Application Server start (6515124)

Description

The bundled Java DB database is not automatically restarted after a host system or Solaris zone reboot, or an Application Server start. This is not a bug, but expected behavior for any bundled or third-party application. The problem is that the Java DB must be started before the Application Server instance.

Solution

After rebooting the host machine or Solaris zone, be sure to start the Java DB *before* starting Application Server; for example:

```
/opt/SUNWappserver/appserver/bin/asadmin start-database
```

Refer to “Application Server Administration Tools” in [Sun Java System Application Server 9.1 Quick Start Guide](#) in the [Sun Java System Application Server 9.1 Quick Start Guide](#) for more information about asadmin command options.

Documentation

This section describes known documentation issues and associated solutions.

Javadoc Inconsistencies (various IDs)

The Javadoc for several AMX interfaces and methods is either missing or incorrect:

- Getter methods for NumConnAcquired and NumConnReleased statistics are missing from ConnectorConnectionPoolStats and AltJDBCCConnectionPoolStats. These getter methods will be added in a future release as getNumConnAcquired() and getNumConnReleased().
- Calling the following methods in EJBCacheStats will throw an exception: getPassivationSuccesses(), getExpiredSessionsRemoved(), getPassivationErrors(), getPassivations(). This will be fixed in a future release.
- The AMX MBeans may require several seconds after server startup before they are all registered and available for use. A future release will make it possible to determine when the AMX MBeans are fully loaded.
- The constant XTypes.CONNNECTOR_CONNECTION_POOL_MONITOR is misspelled ("NNN"). This will be corrected in a future release.

Bundled ANT throws java.lang.NoClassDefFoundError (6265624)

Description

The following exception is thrown in thread "main" java.lang.NoClassDefFoundError: org/apache/tools/ant/launch/Launcher.

Solution

Use the bundled ANT for things outside the Application Server is not recommended.

Online help does not describe how to create a JMS physical destination for a specific server instance or cluster. (6808339)

Use the following procedure to create configuration-specific JMS physical destinations:

1. Log on to the Admin Console.
2. In the left-hand pane, click Standalone Instances > *instance_name*. If you are using clusters, click Clusters > *cluster_name*.
3. Click the JMS Physical Destinations tab.
4. Click New.
5. Create a Queue and Topic.

High Availability

This section describes known high availability database (HADB) issues and associated solutions.

HADB Configuration with Double Networks (no ID)

HADB configured with double networks on two subnets works properly on Solaris SPARC. However, due to problems in the operating system or network drivers on some hardware

platforms, it has been observed that Solaris x86 and Linux platforms do not always handle double networks properly. This causes the following problems with HADB:

- On Linux, some of the HADB processes are blocked when sending messages. This causes HADB node restarts and network partitioning.
- On Solaris x86, some problems may arise after a network failure that prevent switching to the other network interface. This does not happen all the time, so it is still better to have two networks than one. These problems are partially solved in Solaris 10.
- Trunking is not supported.
- HADB does not support double networks on Windows 2003 (ID 5103186).

HADB Database Creation Fails (no ID)

Description

Creating a new database may fail with the following error, stating that too few shared memory segments are available:

```
HADB-E-21054: System resource is unavailable: HADB-S-05512: Attaching shared memory segment with key "xxxxx" failed, OS status=24 OS error message: Too many open files.
```

Solution

Verify that shared memory is configured and the configuration is working. In particular, on Solaris 8, inspect the file `/etc/system`, and check that the value of the variable `shmsys:shminfo_shmseg` is at least six times the number of nodes per host.

`hadbm set` **does not check resource availability (disk and memory space) (5091280)**

Description

When increasing device or buffer sizes using `hadbm set`, the management system checks resource availability when creating databases or adding nodes, but does not check if there are sufficient resources available when device or main-memory buffer sizes are changed.

Solution

Verify that there is enough free disk/memory space on all hosts before increasing any of the `devicesize` or `buffersize` configuration attributes.

Heterogeneous paths for packagepath not supported (5091349)

Description

It is not possible to register the same software package with the same name with different locations at different hosts; for example:

```

hadbm registerpackage test --packagepath=/var/install1 --hosts europa11
Package successfully registered.
hadbm registerpackage test --packagepath=/var/install2 --hosts europa12
hadbm:Error 22171: A software package has already been registered with
the package name test.

```

Solution

HADB does not support heterogeneous paths across nodes in a database cluster. Make sure that the HADB server installation directory (`--packagepath`) is the same across all participating hosts.

`createdomain` may fail (6173886, 6253132)

Description

If running the management agent on a host with multiple network interfaces, the `createdomain` command may fail if not all network interfaces are on the same subnet:

```

hadbm:Error 22020: The management agents could not establish a
domain, please check that the hosts can communicate with UDP multicast.

```

The management agents will (if not configured otherwise) use the "first" interface for UDP multicasts ("first" as defined by the result from `java.net.NetworkInterface.getNetworkInterfaces()`).

Solution

The best solution is to tell the management agent which subnet to use (set `ma.server.mainternal.interfaces` in the configuration file, e.g., `ma.server.mainternal.interfaces=10.11.100.0`). Alternatively one may configure the router between the subnets to route multicast packets (the management agent uses multicast address 228.8.8.8).

Before retrying with a new configuration of the management agents, you may have to clean up the management agent repository. Stop all agents in the domain, and delete all files and directories in the repository directory (identified by `repository.dr.path` in the management agent configuration file). This must be done on all hosts before restarting the agents with a new configuration file.

Starting, stopping, and reconfiguring HADB may fail or hang (6230792, 6230415)

Description

On Solaris 10 Opteron, starting, stopping or reconfiguring HADB using the `hadbm` command may fail or hang with one of the following errors:

```
hadbm:Error 22009: The command issued had no progress in the last
300 seconds.
HADB-E-21070: The operation did not complete within the time limit,
but has not been cancelled and may complete at a later time.
```

This may happen if there are inconsistencies reading/writing to a file (nomandev) which the `clu_noman_srv` process uses. This problem can be detected by looking for the following messages in the HADB history files:

```
n:3 NSUP INF 2005-02-11 18:00:33.844 p:731 Child process noman3 733
does not respond.
n:3 NSUP INF 2005-02-11 18:00:33.844 p:731 Have not heard from it in
104.537454 sec.
n:3 NSUP INF 2005-02-11 18:00:33.844 p:731 Child process noman3 733
did not start.
```

Solution

The following workaround is unverified, as the problem has not been reproduced manually. However, running this command for the affected node should solve the problem.

```
hadbm restartnode --level=clear nodeno dbname
```

Note that all devices for the node will be reinitialized. You may have to stop the node before reinitializing it.

The management agent terminates with the exception "IPv6_MULTICAST_IF failed" (6232140)

Description

When starting on a host running Solaris 8 with several NIC cards installed, if there is a mixture of cards with IPv6 and IPv4 enabled, the management agent may terminate with the exception "IPv6_MULTICAST_IF failed."

Solution

Set the environment variable `JAVA_OPTIONS` to `-Djava.net.preferIPv4Stack=true`; for example:

```
export JAVA_OPTIONS="-Djava.net.preferIPv4Stack=true"
```

Alternatively, use Solaris 9 or later, which do not exhibit this problem.

clu_trans_srv cannot be interrupted (6249685)**Description**

There is a bug in the 64-bit version of Red Hat Enterprise Linux 3.0 that makes the `clu_trans_srv` process end up in an uninterruptible mode when performing asynchronous I/O. This means that `kill -9` does not work and the operating system must be rebooted.

Solution

Use a 32-bit version of Red Hat Enterprise Linux 3.0.

hadbm does not support passwords containing capital letters (6262824)**Description**

Capital letters in passwords are converted to lowercase when the password is stored in `hadb`.

Solution

Do not use passwords containing capital letters.

Downgrading from HADB Version 4.4.2.5 to HADB Version 4.4.1.7 causes ma to fail with different error codes (6265419)**Description**

When downgrading to a previous HADB version, the management agent may fail with different error codes.

Solution

It is possible to downgrade the HADB database, however the management agent cannot be downgraded if there changes have been made in the repository objects. After a downgrade, you must keep use the management agent from the latest HADB version.

Install/removal and symlink preservation (6271063)**Description**

Regarding install/removal of HADB c package (Solaris: `SUNWhadb`, Linux: `sun-hadb-c`) version `<m.n.u-p>`, the symlink `/opt/SUNWhadb/<m>` is never touched once it exists. Thus, it is possible that an orphaned symlink will exist.

Solution

Delete the symlink before install or after uninstall unless in use.

Management agents in global and local zones may interfere (6273681)**Description**

On Solaris 10, stopping a management agent by using the `ma-initd` script in a global zone stops the management agent in the local zone as well.

Solution

Do not install the management agent both in the global and local zone.

hadbm/ma should give a better error message when a session object has timed out and deleted at MA (6275103)**Description**

Sometimes, a resource contention problem on the server may cause a management client to become disconnected. When reconnecting, a misleading error message "hadbm:Error 22184: A password is required to connect to the management agent" may be returned.

Solution

Sometimes, a resource contention problem on the server may cause a management client to become disconnected. When reconnecting, a misleading error message "hadbm:Error 22184: A password is required to connect to the management agent" may be returned.

Check if there is a resource problem on the server, take proper action (e.g., add more resources), and retry the operation.

Non-root users cannot manage HADB (6275319)**Description**

Installing with Java Enterprise System (as root) does not permit non-root users to manage HADB.

Solution

Always login as root to manage HADB.

The Management Agent should not use special-use interfaces (6293912)**Description**

Special use interfaces with IP addresses like `0.0.0.0` should not be registered as valid interfaces to be used for HADB nodes in the Management Agent. Registering such interfaces may cause problems if HADB nodes are set up on these interfaces by means of a user issuing a `hadbm create` command using host names instead of IP addresses. The nodes will then be unable to communicate, causing the `create` command to hang.

Solution

When using `hadbm create` on hosts with multiple interfaces, always specify the IP addresses explicitly using DDN notation.

Reassembly failures on Windows (6291562)

Description

On the Windows platform, with certain configurations and loads, there may be a large number of reassembly failures in the operating system. The problem has been seen with configurations of more than twenty nodes when running several table scans (`select *`) in parallel. The symptoms may be that transactions abort frequently, repair or recovery may take a long time to complete, and there may be frequent timeouts in various parts of the system.

Solution

To fix the problem, the Windows registry variable `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters` can be set to a value higher than the default `100`. It is recommended that you increase this value to `0x1000 (4096)`. For more information, see article [811003](http://support.microsoft.com/default.aspx?scid=kb;en-us;811003) (<http://support.microsoft.com/default.aspx?scid=kb;en-us;811003>) from the Microsoft support pages.

Session state not maintained if the browser has another cookie with / path (6553415)

Description

Cookies with a path equal to `/` interfere with the cookies of a highly available web application deployed at a context root other than `/` that uses in-memory replication as its persistence type, making it impossible for the highly available web application to maintain any HTTP session state. One common scenario where this may happen is when using the same browser to access both the Admin GUI (which is deployed at `/`) and the highly available web application.

Solution

Access the web application deployed at `/` from a different browser.

LB does not work with IIS 6; SASL32.DLL and ZLIB.DLL missing under <appserver-install>/lib (6572184)

Description

`SASL32.DLL` and `ZLIB.DLL` are required files for Load Balancer to work with Windows IIS 6. These files are currently not available under `<appserver-install>/lib`.

Solution

Copy the two DLL files manually to `<appserver-install>/lib`. These files can be downloaded from:

`http://download.java.net/javaee5/external/<OS>/aslb/jars/aslb-9.1-MS4-b5.jar`

Where `<OS>` represents the desired platform, and can be one of the following values:

- SunOS
- SunOS_X86
- Linux
- WINNT

DAS creation/startup and HA package propagation issues in Global Zone (6573511)

Description

Two issues arise when installing or uninstalling Application Server with High Availability packages in a Global Zone:

1. HA packages get installed in all zones, which may not be desirable.
2. When uninstalling, HA, MQ, JDK packages get removed from all zones, which may not be desirable.

This problem does not occur when installing or uninstalling from a root local zone.

Solution

Perform installation and uninstallations from a local root zone rather than a global zone.

Highly available webapps deployed at “/” unable to resume in-memory replicated HTTP sessions (Issue Tracker 2972)

Description

Highly available web applications deployed at “/” are unable to maintain any HTTP sessions when using in-memory replication as their persistence type.

Solution

Deploy highly available web applications that use in-memory replication as their persistence type to a context root other than “/”. If you want to make such a web application available at “/”, you may designate it as the default-web-module of the virtual server to which the web application has been deployed.

AS LB installer did not put /usr/lib/mps path in apachectl LD_LIBRARY_PATH, can not start Apache SSL (6591878)

Description

During Application Server Load Balancer installation for Apache on Solaris, the installer updates LD_LIBRARY_PATH in the apachectl script. However, the installer does not correctly write the /usr/lib/mps path. On Solaris, the Apache security instance will not start without this path in LD_LIBRARY_PATH.

Solution

This issue exists only on Solaris platforms. To work around the issue, add /opt/SUNWappserver/appserver/lib/lbplugin/lib to your LD_LIBRARY_PATH.

Enable/disable LB for an instance/cluster should show correct status (6595113)

Description

The *Enable LoadBalance* button is always enabled on the Clustered/Instance general page, regardless of what is saved in domain.xml.

Solution

- For clustered instances, select *Instances* tab, and then click the *Quiesce* action from the table pull-down.
- For standalone instances, make sure the instance is running, and then click the *Quiesce* button on instance General screen.

AS9.1 EE IFR b58f/JES5 UR1. Can not install Registry Server, because “incomplete” HA was detected. (6602508)

Description

(*Solaris only*) After installing Application Server 9.1 on SPARC Solaris 10 with HADB, you may receive the following error after starting Application Server and then attempting to install JES 5 UR1 with Registry Server:

```
Dependency Error: Installation can not proceed because the version of HA
Session Store 4.4.3 detected on this host is incomplete , and a compatible
version is required by Service Registry Deployment Support.
```

Solution

It is not possible to install Registry Server from JES 5 UR1 with Application Server 9.1 IFR on Solaris machines. The Registry Server packages have to be installed manually using the pkgadd command from the following JES5 UR1 distribution directory:

```
<path>/<OS>/Products/registry-svr/Packages
```

IE 6.0 browser specific: Exporting load balancer configuration file throws error (6516068)

Description

(*Internet Explorer 6 only*) When attempting to export the Load Balancer configuration file (`loadbalancer.xml`) from Internet Explorer 6, the browser displays an error message saying that the `sun-loadbalancer_1_2.dtd` DTD file cannot be located.

Solution

To save the file, use the following workaround:

1. Click *Export* on the Load Balancer page in Internet Explorer.
The “XML page cannot be displayed” message is displayed.
2. Click the error frame, and then choose *File->Save As* from the Internet Explorer.
3. Save the `loadbalancer.xml` file to the directory of your choice.

Installation

This section describes known installation issues and associated solutions.

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

Description

This problem has been observed on several Linux systems. It is most common on Java Desktop System 2 but has also been observed on Linux Red Hat 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-ee/about.html
```

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

On Windows, the `imq` directory needs to be created during installation (6199697)

Description

On Windows, immediately after installing Application Server Enterprise Edition, the Message Queue broker fails on startup with a message saying the directory `drive:\as\domains\domain1\imq` does not exist.

Note that if the broker is started after starting `domain1`, the directory will be created by the Application Server and the problem will not occur.

Solution

1. Create the `var_home_dir_location` before creating the broker:

```
$imqbrokerd -varhome var_home_dir_location
```

For example:

```
$imqbrokerd -varhome D:\as\domains\domain1\imq
```

Spurious warning while installing SDK on Windows Vista (6533646)

Description

When installing the bundled SDK on Windows Vista, you may encounter the error “Unsupported Installation Platform Detected.” However, the installation succeeds without any issues.

Solution

This is not really a problem. Application Server runs on Windows Vista, and this erroneous message will be removed in future versions of the product.

Uninstall does not update `productregistry` file properly; unable to use silent mode to install (6571598)

Description

If the Application Server `productregistry` file contains shared component configurations, an Application Server uninstallation procedure does not update the `productregistry` file correctly, and you will not be able to use silent mode in a subsequent installation unless the `productregistry` file is renamed or removed. Leaving the shared components entries in the `productregistry` file intact is by design, but it leads to confusion with subsequent silent installs.

Solution

After a successful uninstallation is reported back through uninstall log files, delete the product registry file prior to running a subsequent installation. To verify that a previous uninstallation has completed successfully, look for a `appserv_uninstall.class` file in `<install_dir>`. This file will *not* be present if the uninstallation was successful.

Note – Do not delete the product registry if the uninstallation was not successful.

The product registry file is located in `/var/sadm/install` on Solaris and `/var/tmp` on Linux.

IFR. Was not able to install AS in the sparse local zone, MQ packages issue. (6555578)

Description

When installing Application Server in a sparse local zone, the installation fails if Message Queue (MQ) is not installed first. The installer attempts to install MQ, and then the whole installation fails.

Solution

MQ must be manually installed in the global zone before installing Application Server in a sparse local zone. There are two work-arounds for this issue:

1. Install MQ 4.1 manually in the global zone from the same media on which Application Server 9.1 IFR installation is located to get the latest MQ packages.
 - a. Use the installer that corresponds to your platform:

```
mq4_1-installer-SunOS.zip
mq4_1-installer-SunOS_X86.zip
mq4_1-installer-Linux_X86.zip
mq4_1-installer-WINNT.zip
```
 - b. Unzip the bits and run the installer.
The installer will be in the `mq4_1-installer` directory.
2. Install any component of IFR installation in global zone. This action would check the version of MQ in GZ and if required upgrade it to the one bundled in Application Server 9.1 IFR. Even Selecting and Installing the Sample Applications component upgrades MQ to IFR version.
 - a. Run the Application Server installation in the global zone, but select only the sample components.
The sample component installation also installs MQ and Application Server shared components in all zones.
 - b. Run the Application Server installation again, this time in the local sparse zone.
Installation should complete without any problems.

IFR Installation - console prompt has to be removed: "Do you want to upgrade from previous Application?" (6592454)

Description

When running the Application Server 9.1 IFR installer with the `-console` option (command-line mode), you are prompted:

```
Do you want to upgrade from previous Application Server version?
```

Unfortunately, the IFR installer does not support such upgrades, and so this prompt is erroneous. If you answer yes to the prompt, the installation proceeds normally, but no indication that a complete installation was performed, rather than an upgrade.

Solution

Use the upgrade tool if you want to upgrade your Application Server installation.

Java EE Tutorial

To run the Java EE 5 Tutorial on the Sun Java System Application Server 9.1 perform these tasks:

- When you edit the file `examples/common/build.properties` as described in the "About the Examples" section of the "About this Tutorial" chapter, also change port 4848 to 4849.

Note – The default admin port in Application Server 9.1 is again 4848. See ["Default ports changing in each AS major release \(6566481\)" on page 32](#) for more information.

- When using Deploytool, add the server `localhost:4849` before deploying an example.
- When using the Administration Console to create any resource, use the Targets tab to specify the server as the target. If you use the command line or an asant target, the server is the default target, no further action is required.

MDB failure in Java EE Tutorial example (6591307)

Description

In Chapter 32, "Java EE Examples Using the JMS API," in *The Java EE 5 Tutorial*, "An Application Example That Consumes Messages from a Remote Server" in *The Java EE 5 Tutorial*, this example no longer works. The MDB fails to receive the message. The other two examples that send messages between two systems still work correctly ("Running JMS Client Programs on Multiple Systems" in *The Java EE 5 Tutorial* and "An Application Example That Deploys a Message-Driven Bean on Two Servers" in *The Java EE 5 Tutorial*).

Solution

Will be fixed in a later Application Server build.

Java Persistence

TopLink expects my Collection field/property to be cloneable (Issue Tracker 556)

Description

If the `java.util.Arrays.asList()` API is used to convert an `Object[]` to `Collection`, the JDK returns an implementation of `java.util.ArrayList` that is not cloneable. This results in the following exception:

```
The method invocation of the method [protected native java.lang.Object
java.lang.Object.clone() throws java.lang.CloneNotSupportedException] on the object
[[pkg.A id = xxx]], of class [class java.util.Arrays$ArrayList], triggered an
exception. Internal Exception: java.lang.reflect.InvocationTargetException Target
Invocation Exception: java.lang.CloneNotSupportedException:
java.util.Arrays$ArrayList
```

This issue is tracked at https://glassfish.dev.java.net/issues/show_bug.cgi?id=556.

Solution

Create another collection using its constructor; for example:

```
myCollection = new ArrayList(java.util.Arrays.asList(a))
```

Lifecycle Management

This section describes known lifecycle management issues and associated solutions.

Setting ejb-timer-service property minimum-delivery-interval to 9000, an attempt to set the ejb-timer-service property redelivery-interval-in-millis to 7000 causes the set command to fail (6193449)

Description

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

```
[echo] Doing admin task set
[exec] [Attribute(id=redelivery-interval-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-millis` 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.

Error thrown when list JMS physical destinations within non-DAS config (6532532)

Description

If you are trying to view the JMS Physical Destinations using the default-config, you will see an error message.

Solution

This is expected behavior. In Application Server 9.1, `default-config` is a template of configuration information and hence JMS operations (such as `list` and `create`) cannot be executed for the `default-config`. These JMS operations can, however, be executed for the configurations of your cluster or standalone instances.

Win2003 only: Non-paged pool leak memory, breaking tcp stack and richaccess test (6575349)

Description

(Windows 2003 only) There are memory leaks on Windows 2003 systems when performing rich access functions. The problem occurs because the Win32 nonpaged pool keeps growing, eventually bringing down the entire TCP/IP stack. Once the failure happens, the TCP/IP stack is left in an recoverable state, and the only way restore it is by rebooting the Windows 2003 system.

Workaround

There are two workarounds to this issue.

1. Use Grizzly blocking mode by configuring the `domain.xml http-listener` attribute, `blocking-enabled="true"` or add the following `http-listener` property:

```
<property name="blocking" value="true"/>
```

2. Use Windows Vista or Windows XP.

Logging

This section describes known logging issues and solutions.

Setting debug statement for `access, failure` causes hang in Application Server startup (6180095)

Description

Setting the `java.security.debug` option for the JVM will cause the server instance startup to freeze with a deadlock; for example, setting the following in `domain.xml` causes the problem:

```
<jvm-options>-Djava.security.debug=access,failure</jvm-options>
```

Solution

None at this time. Please avoid setting this flag.

Message Queue

This section describes known Java message queue issues and associated solutions.

JMS reconnection does not successfully complete in certain cases that are timing dependent (6173308, 6189645, 6198481, 6199510, 6208728)

Description

Failures to reconnect in timing-dependent scenarios can be caused by several problems.

Solution

You can work around these problems by:

- Restarting the brokers involved
- Restarting the Application Server instances involved

MQ broker fails to start with cluster profile on Linux (6524871)

Description

After creating a domain with a cluster profile on a Linux system, you may encounter a `java.lang.OutOfMemoryError: Java heap space` error, and the server instance may fail to restart because the MQ broker does not start. The system never recovers after this condition. The problem is a misconfigured `/etc/hosts` file; specifically, the server host name is pointing to the loopback address `127.0.0.1`.

Solution

By design, an MQ broker cluster cannot start with the network device configured to point to the loopback address. This is not a bug. The solution is to make sure that the `/etc/hosts` file for the Application Server host does not point to `127.0.0.1`.

Monitoring

This section describes known monitoring issues and associated solutions.

Some of the HTTP Service monitoring statistics do not present useful information and should be ignored (6174518)

Description

When viewing the monitoring statistics of some elements of the HTTP Service, some values presented do not correspond to current values or are always 0. Specifically, the following HTTP Service statistics do not present information applicable to the Application Server, and should be ignored:

- `http-service`
 - `load1MinuteAverage`
 - `load5MinuteAverage`
 - `load15MinuteAverage`
 - `rateBytesTransmitted`
 - `rateBytesReceived`
- `pwc-thread-pool` (the element)

Solution

These monitors will be removed in future releases and replaced with more appropriate information.

Open JNDI Browsing from Admin UI dumps a huge amount of exceptions in the `server.log` (6591734)

Description

Many exceptions are thrown when the JNDI browser is opened from the Admin GUI.

Solution

None at this time.

Samples

This section describes known and associated solutions related to the sample code included with the Application Server 9.1 product.

Documentation does not explicitly state that you need to create JMS resources Documentation does not explicitly state that you need to create JMS resources (6198003)

Description

Documentation does not explicitly state that you need to create JMS resources before running the MQ Failover Sample Application following the `asadmin` deploy instructions.

The error thrown is as follows:

```
/opt/SUNWappserver/domains/domain1/config/sun-acc.xml -name
MQFailoverTestClient -textauth -user j2ee -password j2ee
Nov 18, 2004 10:50:17 PM com.sun.enterprise.naming.NamingManagerImpl
bindObjects
SEVERE: NAM0006: JMS Destination object not found: jms/durable/TopicA
Nov 18, 2004 10:50:18 PM com.sun.enterprise.naming.NamingManagerImpl
bindObjects
SEVERE: javax.naming.NameNotFoundException
javax.naming.NameNotFoundException
```

The documentation does not explicitly state that JMS resources must be manually created if manual deployment is done using `asadmin` deploy commands, and that the provided ant targets to deploy the sample application should be used.

Solution

Use the `asant` deploy target for the `build.xml` script, which creates the required JMS resources to run the application.

On Linux, a runtime error is displayed during certificate creation in web services/security samples (6198239)

Description

When deploying the *install_dir*/samples/webservices/security sample (basicSSL) on Linux, the certificate is not created and an error similar to the following is thrown:

```
generate_certs: [echo] ***Exporting certificate from NSS database
[exec] Result: 1 [echo] ***Generating Java Keystore from generated
certificate [exec] keytool error: java.lang.Exception: Input not an
X.509 certificate [exec] Result: 1 [echo] ***Generating Java trust
store from generated certificate [exec] keytool error: java.lang.
Exception: Input not an X.509 certificate [exec] Result: 1
.
.
.
generate_certs: [echo] ***Exporting server certificate from NSS database to
a PKCS12 certificate file [exec] /opt/sun/appserver/lib/pk12util: /usr/lib/
libnss3.so: version 'NSS_3.9' not found (required by /opt/sun/appserver/lib/
pk12util) [exec] /opt/sun/appserver/lib/pk12util: /usr/lib/libnss3.so:
version 'NSS_3.6' not found (required by /opt/sun/appserver/lib/pk12util)
[exec] /opt/sun/appserver/lib/pk12util: /usr/lib/libnss3.so: version
'NSS_3.7' not found (required by /opt/sun/appserver/lib/pk12util) [exec]
Result: 1
```

The problem is that NSS libraries are in different locations on Linux installations than on Solaris installations. You need to make sure that the LD_LIBRARY_PATH points to the proper NSS libraries when deploying on Linux. Either set LD_LIBRARY_PATH in your environment, or set it in the *install_dir*/bin/asant shell wrapper script.

Solution

Do one of the following:

- Set LD_LIBRARY_PATH=/opt/sun/private/lib.
- Add to the following line to the *install_dir*/bin/asant script:
LD_LIBRARY_PATH=\$AS_NSS:\$LD_LIBRARY_PATH;export LD_LIBRARY_PATH

After upgrade AS9.1 samples and JES5 portal samples compete on derby port 1527 (6574563)

Description

On Windows, after upgrading to Application Server 9.1, the samples and JES5 portal samples compete on Derby port 1527. Specifically, Application Server 9.1 automatically starts JavaDB on port 0.0.0.0:1527 with APP:APP, however the JES5 Portal JavaDB wants to bind to hostnameIP:1527 with portal:portal.

This bug describes an issue that was already seen for JES 5, Bug 6472173. The workaround for bug 6472173 is documented in the [Sun Java Enterprise System 5 Installation Guide for Microsoft Windows](#).

Solution

Start the Derby database using the following command:

```
<JES installation dir>\appserver\bin\asadmin start-database --dbhome <JES installation dir>\portal\data\derby
```

Security

This section describes known issues and associated solutions related to Application Server and web application security and certificates.

SSL termination is not working (6269102)

Description

SSL termination is not working; when Load Balancer (Hardware) is configured for SSL termination, the Application Server changes the protocol from https to http during redirection.

Solution

Add a software load balancer between the hardware load balancer and the Application Server.

Socket connection leak with SSL (6492477)

Description

Because of a JVM bug, there is a leak issue with some JDK versions when security-enabled is set to true on an HTTP listener. Specifically, the steps to reproduce this bug are as follows:

1. Set security-enabled to true on the HTTP listener:

```
<http-listener acceptor-threads="1" address="0.0.0.0"
blocking-enabled="false" default-virtual-server="server" enabled="true"
family="inet" id=" http-listener-1" port="8080" security-enabled="true"
server-name="" xpowered-by="true">
```

2. Comment out stopping domain at the end of quicklook tests.
3. Run quicklook tests.
4. Check socket usage:

```
netstat -an | grep 8080
```

The following are shown to be in use:

*.8080	*,*	0	0 49152	0 LISTEN
*.8080	*,*	0	0 49152	0 BOUND

This issue is tracked on the Glassfish site at https://glassfish.dev.java.net/issues/show_bug.cgi?id=849.

Solution

Upgrade to the latest JDK version.

Upgrade Utility

This section describes known Upgrade utility issues and associated solutions.

Domains created in custom-path other than *install_dir*/domains directory are not upgraded directly (6165528)

Description

Domains created in custom-path other than *install_dir*/domains directory are not upgraded directly while upgrading from Application Server Enterprise Edition 8 to Application Server Enterprise Edition 8.1.

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.

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

Description

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

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-ee/about.html
```

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

Self-signed certificate is not trusted during and after upgrade from 8.0 Platform Edition (PE) to 8.1 Enterprise Edition (EE) UR2 (6296105)

Solution

Remove the following entries from the `target.domain.xml` (after the upgrade) and restart the server:

```
<jvm-options>-Djavax.net.ssl.keyStore=${com.sun.aas.instanceRoot}
/config/keystore.jks</jvm-options>-
<jvm-options>Djavax.net.ssl.trustStore=${com.sun.aas.instanceRoot}
/config/cacerts.jks</jvm-options>
```

Upgrade tool overwrites the content of a modified `index.html` of an instance (member of a cluster) (6386451)

Description

The upgrade tool overwrites any existing `index.html` file for any server instance.

Solution

Back up your existing `index.html` files before running the upgrade tool, and then restore those files later.

(sbs-manual, sbs-installer) throws Server Instance server does not have a system connector named null (6545145)

Description

When upgrading from Application Server 8.0PE to 9.1, an error is thrown saying the server does not have system connector named `null`, and invalid user information as seen in `sbs-manual`. Even after changing the hardcoded values, the same error message is seen. This occurs because the `domain.xml` has changed between 8.0 and 9.1.

Solution

You can only encounter this bug while upgrading from a 8.0 PE to 9.1. The workaround is to upgrade to either 8.1, 8.2, or 9.0 and then upgrade to 9.1.

Different domains are lost during upgrade when different build combinations are used (6546130)

Description

When performing an inplace upgrade, in cases where there are multiple domains in the source, the installer invokes upgrade tool even though the process is killed. This happens when it is invoked in GUI mode.

Solution

1. Install inplace in the CLI mode, and exit when the installer prompts you to select the upgrade tool at the end of installation process. This does not delete any of the domains present in the domains directory. Upgrade tool should be manually invoked from the bin directory.
2. When installing inplace in GUI mode, make a backup of the domains present in the domains root to prevent losing any domains in the process. At the end of the installation process, exit when the installer prompts you to invoke the upgrade tool. Copy any backed up domains into the domains directory if they have been lost. Launch upgrade tool manually to do an upgrade.

Solaris: pre-filled Master password in upgrade Tool is from AS9.1 not from AS8.2 (6565825)

Description

When upgrading from AS 8.2 to 9.1, the master password from the 8.2 installation is not inherited in the 9.1 installation. This subsequently causes an authentication error at the next admin login.

Solution

The default admin password in Application Server 9.1 is `changeit`. To avoid problems when logging in to the 9.1 server after upgrading from 8.2, do one of the three following things:

- Change the 8.2 admin password to `changeit` before performing the upgrade.
- Do not accept the default admin password during the upgrade process, but instead explicitly enter the password you want to use.
- Log in to 9.1 with the default password, and then immediately change it.

AS9.1 IFR upgrade tool doesn't migrate JES5U1 MQ store to the new var home (6573635)

Description

The upgrade tool does not deal with upgrading databases or database tables in any form, nor will it ever support this. The resource references configurations are transferred and the Application Server should continue to work with the original database and tables. If you want to change the database or transfer database tables, use the tools that work with the databases in use.

Solution

Perform the following steps to migrate the MQ store:

Note – Perform the following steps **AFTER** AS 8.2 has been shut down and **AFTER** the AS9.1 upgrade tool is run but **BEFORE** AS9.1 is started for the **FIRST** time. If you have already started AS 9.1 after the IFR install/upgrade, then **DO NOT** perform these steps as they will potentially destabilize the MQ message store.

1. Copy the entire domains/domain1/imq subdirectory from the AS 8.x domains directory to the AS 9.1 domains directory.
2. Ensure that the ownership of the directory and files is the same as the user that is going to be running the Application Server.

After performing the above steps, Application Server 9.1 can be started and the MQ store in the Application Server 9.1. domains directory will be migrated from its JES5 U1 format to MQ 4.1 format. Note that the original JES5 U1 MQ store under AS 8.2 is preserved and unmodified by this procedure or by MQ4.1 upon startup by AS 9.1

Upgrade JES5 (AS 8.2) to AS 9.1. PS Community sample is broken, many javax.faces errors (6576700)

Description

When upgrading from JES5 (Application Server 8.2) to Application Server 9.1, the Portal Server Community sample no longer works, and throws many javax.faces.application.ApplicationFactory errors.

Solution

Upgrading from Application Server 8.2 to 9.1 is not supported if Application Server 8.2 was installed with the JES5 Portal Server. Portal Server needs to be upgraded to Java ES 5 Update 1 prior the Application Server upgrade to 9.1.

AS91 IFR b58, Linux: /usr/jdk/entsys-j2se link is not updated after installing new Java via Installer (6591697)

Description

When upgrading from Application Server 8.2 to 9.1 with the IFR installer on Linux platforms, selecting the *Install JDK* option, but after successful completion of the installation, most of the JES components stop working.

Solution

This issue only affects IFR installation of Application Server 9.1 in Linux platforms, and only when the *Install JDK* option is selected. To work around this issue, immediately after installation manually link /usr/jdk/entsys-j2se to the /usr/java/jdk1.5.0_12 directory.

AS9.1 IFR Win in-place backup is not integrated with asupdate.bat form values checking - wrong path (6596412)

Description

When performing an Application Server 9.1 IFR upgrade on Windows, in-place backup is not correctly integrated with asupdate.bat form values. Specifically, if you enter incorrect information in an ASupdate.bat GUI screen and then click *Next*, the upgrade installer tries to detect if it is an in-place upgrade. If yes, domain1 is moved to a backup directory prior upgrade. As the upgrade proceeds, and an error message is displayed as a result of the incorrect information. When you attempt to correct the error right away, the a path error is thrown because domain1 was already moved.

Solution

Either change the source directory to be the domain1_{timestamp} directory in {current source path}/backup or exit the installer with the *Cancel* button and start again.

AS9.1 IFR Windows, inline upgrade, Domain1 is deleted when asupgrade.bat used with short and long path (6596394)

Description

(Windows only) If an earlier version of Application Server was installed using special characters or DOS-style short names in the program directory path, subsequent in-place upgrades to Application Server 9.1 will fail if those same directory path names are used.

For example, if Application Server 8.2 was installed in either:

```
C:\Program Files (x86)\dirs\appserver
c:\progra~2\dirs\appserver
```

Attempts to perform an in-place upgrade to 9.1 will fail because the installer cannot convert the short names or special characters to the required long name format.

Solution

Installing Application Server using a path name that contains special characters or the DOS-style short name truncation (such as `progra~2`) is strongly discouraged because it impedes subsequent upgrade installations. If such an installation exists, either reinstall it using long path names before upgrading, or install the new version of Application Server in an entirely new directory.

AS 9.1 IFR: Web UI Authentication Broken (6569813)

Description

After an Application Server upgrade, the `<jsp:forward>` tag does not work as expected in `Authenticate.jsp`. The `<jsp:forward>` call produces an error in the server logs and a blank page is shown on the WebUI. The problem is that `<jsp:forward>` in `Authenticate.jsp` requires a page attribute like `<jsp:forward page="${redirectPage}"/>`, but the value being passed is a relative path like `/registry/thin/{pagename}.jsp`, which does not work even `Authenticate.jsp` is a pure JSP page.

Solution

After completing the Application Server upgrade, use the `asadmin` tool to run the following commands to set the `<auth-realm>` in `domain.xml`:

1. Go to `<appserver9.1-install-dir>/bin` and run the following command:

```
./asadmin delete-auth-realm --host localhost --port 6489 certificate
```

This removes the old `auth-realm` certificate, if one exists.

2. Run the following command:

```
./asadmin create-auth-realm --terse=false --echo=true --interactive=true \  
--user admin --host localhost --port 6489 --classname \  
com.sun.enterprise.security.auth.realm.certificate.CertificateRealm \  
--property assign-groups=have.client.cert certificate
```

This creates the new `<auth-realm>` with the `assign-groups` property.

3. Stop and restart the Application Server registry domain.

Localized Online Help for asupgrade GUI Does Not Exist (6610170)

Descriptino

When running the `asupgrade` GUI in a language other than English, the online help for the GUI is not localized for the selected non-English language.

Solution

None at the time. Online help is scheduled to be localized in all non-English target languages.

Web Container

This section describes known web container issues and associated solutions.

On Windows, deploying an application using `--precompilejsp=true` can lock JAR files in the application, causing later undeployment or redeployment to fail (5004315)

Description

If you request precompilation of JSPs when you deploy an application on Windows, later attempts to undeploy that application or to redeploy it (or any application with the same module ID) will not work as expected. The problem is that JSP precompilation opens JAR files in your application but does not close them, and Windows prevents the undeployment from deleting those files or the redeployment from overwriting them.

Note that undeployment succeeds to a point, in that the application is logically removed from the Application Server. Also note that no error message is returned by the `asadmin` utility, but the application's directory and the locked jar files remain on the server. The server's log file will contain messages describing the failure to delete the files and the application's directory.

Attempts to redeploy the application after undeploying fail because the server tries to remove the existing files and directory, and these attempts also fail. This can happen if you try to deploy any application that uses the same module ID as the originally deployed application, because the server uses the module ID in choosing a directory name to hold the application's files.

Attempts to redeploy the application without undeploying it first will fail for the same reasons.

Diagnostics

If you attempt to redeploy the application or deploy it after undeploying it, the `asadmin` utility returns an error similar to the one below.

```
An exception occurred while running the command. The exception
message is: CLI171 Command deploy failed : Deploying application in
domain failed; Cannot deploy. Module directory is locked and can't
be deleted.
```

Solution

If you specify `--precompilejsps=false` (the default setting) when you deploy an application, then this problem will not occur. Be aware that the first use of the application will trigger the JSP compilation, so the response time to the first request will be longer than for later requests.

Note also that if you do precompile, you should stop and restart the server before undeploying or redeploying the application. The shutdown frees the locked JAR files so the undeployment or redeployment after the restart can succeed.

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

Description

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.

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 (6184122)

Description

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.

Application Server does not support auth-passsthrough Web Server 6.1 Add-On (6188932)

Description

The Sun Java System Application Server 9.1 adds support for the functionality provided by the auth-passsthrough plugin function available with Sun Java System Application Server Enterprise Edition 7.1. However, in Application Server 9.1, the auth-passsthrough plugin feature is configured differently.

The auth-passsthrough plugin function in Application Server Enterprise Edition 7.1 has been useful in two-tier deployment scenarios, where:

- Application Server instance is protected by a second firewall behind the corporate firewall.
- No client connections are permitted directly to the Application Server instance.

In such network architectures, a client connects to a front-end web server, which has been configured with the service-passsthrough plugin function and forwards HTTP requests to the proxied Application Server instance for processing. The Application Server instance can only receive requests from the web server proxy, but never directly from any client hosts. As a result of this, any applications deployed on the proxied Application Server instance that query for client information, such as the client's IP address, will receive the proxy host IP, since that is the actual originating host of the relayed request.

Solution

In Application Server Enterprise Edition 7.1, the auth-passsthrough plugin function could be configured on the proxied Application Server instance in order to make the remote client's information directly available to any applications deployed on it; as if the proxied Application Server instance had received the request directly, instead of via an intermediate web server running the service-passsthrough plugin.

In Application Server 9.1, the auth-passthrough feature may be enabled by setting the `authPassthroughEnabled` property of the `<http-service>` element in `domain.xml` to `TRUE`, as follows:

```
<property name="authPassthroughEnabled" value="true"/>
```

The same security considerations of the auth-passthrough plugin function in Application Server Enterprise Edition 7.1 also apply to the `authPassthroughEnabled` property in Application Server 9.1. Since `authPassthroughEnabled` makes it possible to override information that may be used for authentication purposes (such as the IP address from which the request originated, or the SSL client certificate), it is essential that only trusted clients or servers be allowed to connect to an Application Server 9.1 instance with `authPassthroughEnabled` set to `TRUE`. As a precautionary measure, it is recommended that only servers behind the corporate firewall should be configured with `authPassthroughEnabled` set to `TRUE`. A server that is accessible through the Internet must never be configured with `authPassthroughEnabled` set to `TRUE`.

Notice that in the scenario where a proxy web server has been configured with the `service-passthrough` plugin and forwards requests to an Application Server 8.1 Update 2 instance with `authPassthroughEnabled` set to `TRUE`, SSL client authentication may be enabled on the web server proxy, and disabled on the proxied Application Server 8.1 Update 2 instance. In this case, the proxied Application Server 8.1 Update 2 instance will still treat the request as though it was authenticated via SSL, and provide the client's SSL certificate to any deployed applications requesting it.

Web Server

AS 9.1 b50e.Linux.Can not start WS after AS LB installation: libjvm.so:cannot open shared (6572654)

Description

This issue only applies if you are using the Sun Java System Web Server with Application Server 9.1 and Load Balancer on a Linux system. In such a case, after installing Application Server and a load balancer, the Web Server may fail to start because `libcui18n.so.2` and `libcuc.so.2` are in conflict. These libraries are present in both `/opt/sun/private/lib` and `/opt/sun/appserver/lib`.

Solution

The correct libraries to use are the ones in `/opt/sun/appserver/lib` because `lbplugin` is built against those libraries. Once you remove the two libraries from `/opt/sun/private/lib`, Web Server should start without error.

Alternatively, if you do not want to delete the libraries from `/opt/sun/private/lib`, you can instead put `/opt/sun/appserver/lib` before `/opt/sun/private/lib` in `LD_LIBRARY_PATH` in the Web Server `startserv` script; that is, replace:

```
# Add instance-specific information to LD_LIBRARY_PATH for Solaris and Linux
LD_LIBRARY_PATH="${SERVER_LIB_PATH}:${SERVER_JVM_LIBPATH}:${LD_LIBRARY_PATH}:/opt/sun/appserver/lib:/opt/sun/appserver/lbplugin/lib"; export LD_LIBRARY_PATH
```

with:

```
# Add instance-specific information to LD_LIBRARY_PATH for Solaris and Linux
LD_LIBRARY_PATH="/opt/sun/appserver/lib:/opt/sun/appserver/lbplugin/lib:${SERVER_LIB_PATH}:${SERVER_JVM_LIBPATH}:${LD_LIBRARY_PATH}"; export LD_LIBRARY_PATH
```

Web Services

This section describes known web container issues and associated solutions.

Ant task `wsimport` fails with Java EE SDK b33d (using JDK 1.6) with `NoClassDefFoundError` (6527842)

Description

You may encounter a problem when running the JAX—WS tests with the JDK 1.6 included with the Java EE SDK b33d. The tests immediately abort with the following message:

```
[wsimport] Exception in thread "main" java.lang.NoClassDefFoundError: \
com/sun/tools/ws/WsImport
```

This error occurs even though the `webservices-tools.jar` does contain `com/sun/tools/ws/WsImport.class`, `com/sun/tools/ws/ant/WsImport.class`, and `com/sun/tools/ws/ant/WsImport2.class`. Moreover, the same test workspace works without problem using the 1.5.0-10 JDK.

Solution

Copy the `webservices-api.jar` to `$JAVA_HOME/jre/lib/endorsed` before running the JAX-WS tests.

publish-to-registry commands fail in IFR EE builds (6602046)

Description

JAXR uses SAAJ to send soap messages to the registry. In the non-IFR case, the SAAJ impl classes are under `lib/webservices-rt.jar`. In the IFR case, the SAAJ classes are still under `lib/webservices-rt.jar`. In addition, `saaj-impl.jar` is located in the `/usr/share/lib` directory. This jar file is picked up by Application Server and has precedence over classes from `webservices-rt.jar`. This jar file does not have the necessary security permissions to send soap messages to the Web services registry. The packaging should be modified to grant permissions to the jars under `/usr/share/lib` directory or not depend on the `/usr/share/lib` jars.

Solution

Add the following to the `server.policy` file:

```
grant codeBase "file:/usr/share/lib/saaj-impl.jar" {  
    permission java.security.AllPermission;  
};
```