Sun Java System Application Server Enterprise Edition 8.2 Release Notes

Sun microsystems

1

Overview

The Sun Java™ System Application Server Enterprise Edition 8.2 product greatly simplifies the task of creating and administering J2EE applications and web services. It provides superior performance, clustering, and high availability features for scalable services that continue to operate despite software and hardware faults.

- "About These Notes" on page 3
- "Release Notes Revision History" on page 4
- "Application Server Documentation Set" on page 4
- "Related Documentation" on page 5
- "Accessibility Features" on page 6
- "Documentation, Support, and Training" on page 6
- "How to Report Problems and Provide Feedback" on page 6
- "Sun Welcomes Your Comments" on page 7

About These Notes

These Release Notes contain important information available at the time of release of Sun Java System Application Server 8.2. Enhancements, known problems, and other late-breaking issues are addressed here. Read this document before you begin using Application Server Enterprise Edition 8.2.

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/1310.3`). Check the Web site prior to installing and setting up your software and then periodically thereafter to view the most up-to-date release notes and product documentation.

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

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

Release Notes Revision History

This section lists the changes that have been made in these release notes after the initial release of the Sun Java SystemEnterprise Edition Application Server 8.2 product.

TABLE 1–1 Release Notes Revision History

Revision Date	Description
October 2006	Initial notes for the Beta release of the Sun Java System Application Server Enterprise Edition 8.2 product.
February 2007	Notes for FCS release of the Sun Java System Application Server Enterprise Edition 8.2 product.
July 2007	Added defect 6396045 to known problems for installation.
August 2007	Changed platform requirements for WebServer to reflect Solaris and Linux platforms supported for Java ES 5.
May 2008	Changed supported Web Servers to 6,0, 6.1, 7.0.
	Added "System Virtualization Support" section.
July 2009	Added defect 6860787 to known problems for administration.
November 2009	Added defect 6898037 to known problems for upgrade.

Application Server Documentation Set

The Application Server documentation set describes deployment planning and system installation. The Uniform Resource Locator (URL) for stand-alone Application Server documentation is `http://docs.sun.com/app/docs/coll/1310.4`. The URL for Sun Java Enterprise System (Java ES) Application Server documentation is `http://docs.sun.com/app/docs/coll/1310.3`. For an introduction to Application Server, refer to the books in the order in which they are listed in the following table.

TABLE 1–2   Books in the Application Server Documentation Set

Book Title	Description
Release Notes	Late-breaking information about the software and the documentation. Includes a comprehensive, table-based summary of the supported hardware, operating system, Java Development Kit (JDK^{TM}), and database drivers.
Quick Start Guide	How to get started with the Application Server product.
Installation Guide	Installing the software and its components.
Deployment Planning Guide	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.
Developer's Guide	Creating and implementing Java 2 Platform, Enterprise Edition (J2EE^{TM} platform) applications intended to run on the Application Server that follow the open Java standards model for J2EE components and APIs. Includes information about developer tools, security, debugging, deployment, and creating lifecycle modules.
J2EE 1.4 Tutorial	Using J2EE 1.4 platform technologies and APIs to develop J2EE applications.
Administration Guide	Configuring, managing, and deploying Application Server subsystems and components from the Administration Console.
High Availability Administration Guide	Post-installation configuration and administration instructions for the high-availability database.
Administration Reference	Editing the Application Server configuration file, `domain.xml`.
Upgrade and Migration Guide	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.
Performance Tuning Guide	Tuning the Application Server to improve performance.
Troubleshooting Guide	Solving Application Server problems.
Error Message Reference	Solving Application Server error messages.
Reference Manual	Utility commands available with the Application Server; written in man page style. Includes the `asadmin` command line interface.

Related Documentation

Application Server can be purchased by itself or as a component of Java ES, a software infrastructure that supports enterprise applications distributed across a network or Internet environment. If you purchased Application Server as a component of Java ES, you should be familiar with the system documentation at `http://docs.sun.com/coll/1286.2`. The URL for all documentation about Java ES and its components is `http://docs.sun.com/prod/entsys.5`.

For other Sun Java System server documentation, go to the following:

- Message Queue documentation
- Directory Server documentation
- Web Server documentation

Additionally, the following resources might be useful:

- The J2EE 1.4 Specifications (`http://java.sun.com/j2ee/1.4/docs/index.html`)
- The J2EE 1.4 Tutorial (`http://java.sun.com/j2ee/1.4/docs/tutorial/doc/index.html`)
- The J2EE Blueprints (`http://java.sun.com/reference/blueprints/index.html`)

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`.

Documentation, Support, and Training

The Sun web site provides information about the following additional resources:

- Documentation (`http://www.sun.com/documentation/`)
- Support (`http://www.sun.com/support/`)
- Training (`http://www.sun.com/training/`)

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`) — A form for submitting feedback on the Application Server product

- J2EE-INTEREST list (`http://archives.java.sun.com/archives/j2ee-interest.html`) — A mailing list for J2EE questions

- Bug database on Java Developer Connection (`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/`) — An interactive message board for sharing knowledge and questions about Java technologies and programming techniques; use the J2EE SDK forum for discussions related to the Sun Java System Application Server Enterprise Edition 8.2 product

- Sun Software Support services (`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 full document title and part number. The part number is a 7-digit or 9-digit number that can be found on the book's title page or in the document's URL. For example, the part number of this book is 819-4728.

2

About Application Server Enterprise Edition 8.2

The Sun Java System Application Server Enterprise Edition 8.2 is a J2EE 1.4 platform-compatible server for the development and deployment of J2EE applications and Java technology-based web services in large-scale production environments.

This chapter includes:

What's New in the 8.2 Release

The Application Server Enterprise Edition 8.2 includes the following enhancements:

- **Improved Administration** — The Application Server supports the remote secure management of complex multi-machine enterprise deployments using either a browser based console or a scriptable command line interface. It also provides a rich JMX based API allowing remote, secure, programmatic access to administrative and monitoring functions.

- **Message Broker** — The Application Server is bundled with an integrated enterprise class message broker that features providing highly available, reliable, high performance, and scalable messaging.

- **Message Queue 3.7** — The Application Server now implements MQ 3.7.

- **Expanded Platform Support** — Additional operating systems, databases, locales, and hardware are supported.

- **Sun Java Enterprise System** — As a key component of the Sun Java Enterprise System, the Application Server is tightly integrated with portal and network identity services.

- **Migration and Upgrade Tools** — These tools enable you to verify J2EE applications for standards conformance and portability, help with migrations from other J2EE Application Servers (JBoss, WebLogic, WebSphere), and aid in upgrading from previous versions of Sun ONE Application Server/ iPlanet Application Server.

- **Java 2 Standard Edition 5.0 Support** — The Application Server supports the Java 2 Standard Edition 5.0, which includes enhanced management and monitoring features and many performance and scalability improvements.

- **Java Web Services Developer Pack 1.6 (JWDSP) Plugin Support** — All JWSDP plugins are now supported. The JWSDP 1.6 can be downloaded for free from `http://java.sun.com/webservices/downloads/1.6/index.html`.

- **Java DB Database Support** — Application Server includes the Java DB database, based on Apache Derby (`http://db.apache.org/derby/`). Backward compatibility with Pointbase database is maintained, but any new databases created on the server will use Java DB by default. After upgrading from Application Server 8.x, existing domains will continue to use PointBase, but any new domain created after the upgrade will use Java DB.

- **JDBC Drivers** — The Application Server is bundled with Sun JDBC drivers.

- **Web Services Security** — These container message security mechanisms implement message-level authentication (for example, XML digital signature and encryption) of SOAP web services invocations using the X509 and username/password profiles of the OASIS WS-Security standard.

- **WS-I Basic Profile 1.1** — As mandated by the J2EE 1.4 specification, this release implements Web Services Interoperability (WS-I) Basic Profile 1.1 to enable interoperability for web services applications.

- **Backend Connectivity with iWay Adapters** — Sun Microsystems now resells and supports twenty-two iWay adapters to key backend systems (SAP, Siebel, Oracle, CICS, and IBM MQ Series) to help you leverage existing IT applications from within the Application Server environment. These adapters support the J2EE Connector Architecture 1.5 specification and Web services (SOAP) standards, and include developer tools to reduce time to connect to backend applications.

- **Latest HADB Management System** — The UNIX™ platforms contain the new high availability database (HADB) management system (HADB version 4.4.3), which includes a Database Server, ODBC 2.5 driver, JDBC 3.0 type 4 driver, `clusql` (an interactive program to enter and execute SQL statements), and a management system. This version eliminates the dependency on SSH/RSH, but requires that the network be configured for UDP multicast. See the *Sun Java System Application Server Enterprise Edition 8.2 High Availability Administration Guide* for the details on HADB requirements and limitations.

- **Solaris 10 Zones Support** – The Application Server can be installed in either a global or non-global Zone on Solaris 10 systems. See the Solaris Zones (`http://www.sun.com/bigadmin/content/zones/`) page for more information about Solaris Zones.

- **Dynamic Content Technology Support Dropped** — Dynamic content technologies, such as CGI-bin and SHTML, are no longer supported.

Hardware and Software Requirements

This section lists the requirements that must be met before installing the Sun Java System Application Server Enterprise Edition 8.2 product.

- "Platform Requirements" on page 11
- "System Virtualization Support" on page 12
- "Important Patch Information" on page 12
- "JDBC Drivers and Databases" on page 12
- "Using the Bundled Java DB Database" on page 13
- "Supported Web Servers" on page 17
- "Browsers" on page 17
- "HADB Requirements and Supported Platforms" on page 18
- "Upgrading the Sun Java System Application Server" on page 19
- "Other Requirements" on page 19

Platform Requirements

The following table lists the operating systems that are supported for Sun Java System Application Server Enterprise Edition 8.2 product. Additionally, the minimum and recommended memory requirements are identified for installing and running the Application Server.

TABLE 2–1   Sun Java System Application Server 8.2 Platform Requirements

Operating System	Minimum Memory	Recommended Memory	Minimum Disk Space	Recommended Disk Space	JVM[1]
Sun Solaris 9, 10 (SPARC) Solaris 9, 10 (x86)	512 MB	1 GB	250 MB free	500 MB free	J2SE_5_08
Sun Java Desktop System	512 MB	1 GB	250 MB free	500 MB free	J2SE_5_08
Redhat Enterprise Linux 3.0 U1, 4.0	512 MB	1 GB	250 MB free	500 MB free	J2SE_5_08
Windows Server 2000 SP4+ Windows 2000 Advanced Server SP4+ Windows Server 2003 Windows XP Pro SP1+	1 GB	2 GB	500 MB free	1 GB free	J2SE_5_08

[1] Only 32–bit (rather than 64–bit) JVMs are supported.

Note – The system requirements listed above for Application Server and those listed for HADB in "HADB Requirements and Supported Platforms" on page 18 are not exactly the same. This is not a documentation error. It is not uncommon to run Application Server and an HADB server on different machines.

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

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

System Virtualization Support

System virtualization is a technology that enables multiple operating system (OS) instances to execute independently on shared hardware. Functionally, software deployed to an OS hosted in a virtualized environment is generally unaware that the underlying platform has been virtualized. Sun performs testing of its Sun Java System products on select system virtualization and OS combinations to help validate that the Sun Java System products continue to function on properly sized and configured virtualized environments as they do on non-virtualized systems. For information about Sun support for Sun Java System products in virtualized environments, see System Virtualization Support in Sun Java System Products.

Important Patch Information

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 (http://sunsolve.sun.com/) on SunSolve.

RedHat Enterprise Linux 3.0 Additional Package Requirements

To run native components of this product, including installer, the following package, which is not part of the standard RedHat Enterprise Linux 3.0 distribution, should be installed: `compat-libstdc++-7.3-2.96.118.i386.rpm`

The package can be downloaded from http://rpm.pbone.net/ index.php3/stat/4/idpl/843376/com/compat-libstdc++-7.3-2.96.118.i386.rpm.html

JDBC Drivers and Databases

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

TABLE 2–2 J2EE-Compatible JDBC Drivers

JDBC Vendor	JDBC Driver Type	Supported Database Server
i-net Software	Type 4	Oracle (R) 8.1.7, 9i, 9.2.0.3+, 10.1.*x*, 10.2.*x*
		Sybase ASE 12.5.
		Microsoft SQL Server 2000 4.0 Service Pack 1
IBM	Type 2	IBM DB2 8.1 Service Pack 3+
Java DB	Type 4	Apache Derby 10.1.3
PointBase	Type 4	PointBase Network Server 5.2
DataDirect	Type 4	Oracle (R) 8.1.7, 9i, 9.2.0.3+, 10.1.*x*, 10.2.*x*
		Sybase ASE 12.5.2
		Microsoft SQL Server
		IBM DB2 8.1 Service Pack 3+
MySQL	Type 4	5.*x*
Sun Java System JDBC Driver for Oracle	Type 4	Oracle (R) 9.2.0.3, 10G
Sun Java System JDBC Driver for DB2	Type 4	IBM DB2 8.1 Service Pack 3+
Sun Java System JDBC Driver for Sybase	Type 4	Sybase ASE 12.5.2
Sun Java System JDBC Driver for Microsoft SQL Server	Type 4	Microsoft SQL Server 2000 4.0 Service Pack 1
Oracle	Type 4, Type 2	Oracle (R) 9.2.0.3, 10G

Using the Bundled Java DB Database

This section provides instructions for using the Java DB database implementation bundled with Application Server 8.2.

-
-
-

Starting and Stopping the Java DB Database

Sun Java System Application Server 8.2 introduces two new `asadmin` commands for starting and stopping the Java DB Network Server.

- The `start-database` command can be used to start an instance of the Java DB network server:

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

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

■ The asadmin stop-database command is used to shut down an instance of the Java DB network server that is running:

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

Java DB Utility Scripts

The Java DB configuration that ships with Application Server 8.2 also includes several useful scripts which can help you use Java DB. The following scripts are available for use in the *<appserver_install_dir>*/derby/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>*/derby **directory.**

2 Unset your CLASSPATH **environment variable.**

3 You can also optionally set the following properties:

a. DERBY_SERVER_HOST **to the host on which the network server will listen.**
Can also be set to `0.0.0.0` to enable all listeners.

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

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

Exporting Tables from Pointbase to Java DB

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

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


PointBase Commander 5.2 ECF build 294 size restricted version EMBEDDED


Interactive SQL command language. SunOS/5.9


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


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


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

The results from executing the unload database command is written in the above example to the file sampledb.sql. The sampledb.sql file contains all of the DDL required to create the necessary tables and indexes. It also contains the DML to insert the data back into the database. The commander command RUN is intended to be used import the data into another Pointbase

database using the script that was generated. Here is an example of what the INSERT statements and associated data look like in the generated file:

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

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

- Removed the phrase Organization Heap from the end of all CREATE Table statements

- Removed the COMMIT command

- Changed the Boolean datatype to be smallint

- Removed all of the INSERT statements and associated data

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

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

Converting Pointbase Java procedures to work with Java DB requires some changes to the Java code as well as to the CREATE PROCEDURE statements. Information on creating Java DB Java

procedures can be found in the Derby Reference manual (`http://db.apache.org/derby/docs/10.1/ref/`). Support for the `Boolean` datatype should be in the next release of Java DB.

Supported Web Servers

This section lists the web servers that are supported for the Sun Java System Application Server Enterprise Edition 8.2.

TABLE 2–3 Supported Web Servers

Web Server	Version	Operating System
Sun Java System Web Server	6,0, 6.1, 7.0	Solaris SPARC 9, 10
		Solaris *x*86 9, 10
		Red Hat Enterprise Linux 3 and 4
Apache Web Server	1.3+, 1.4, 2.0	Solaris SPARC 9, 10
		Solaris *x*86 10
		Red Hat Enterprise Linux 3 and 4
		Windows Server 2003
		Windows 2000 Advanced Server SP4+
		Windows Server 2000 SP4+
		Windows XP Pro SP1+
Microsoft IIS™	5.0+	Windows Server 2003
		Windows 2000 Advanced Server SP4+
		Windows Server 2000 SP4+
		Windows XP Pro SP1+

Browsers

This section lists the browsers that are supported with the Sun Java System Application Server Enterprise Edition 8.2.

TABLE 2–4 Supported Web Browsers

Browser	Version
Mozilla	1.4, 1.5, 1.6, 1.7.*x*
Netscape Navigator	4.79, 6.2, 7.0, 8.*x*
Internet Explorer	5.5 Service Pack 2, 6.0

Browser	Version
Firefox	1.4, 1.5

TABLE 2–4   Supported Web Browsers   *(Continued)*

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.

HADB 4.4.3 is co-packaged with Application ServerEnterprise Edition 8.2

Note – The system requirements listed in "Platform Requirements" on page 11 for Application Server and those listed here for HADB are not exactly the same. This is not a documentation error. It is not uncommon to run Application Server and an HADB server on different machines.

- "Supported Platforms" on page 18
- "HADB Server Host Requirements" on page 18
- "HADB Management Host Requirements" on page 19
- "HADB Client Host Requirements" on page 19

Note – The Java components of the system have been built with JDK 1.4.2_02 and has been tested on JDK 1.5_09.

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 50 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** - 512 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** - 1 GB 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 *x*86.

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

The in-place upgrade from any prior Application Server release is not supported. Refer to the *Application Server Enterprise Edition 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 settings. By default, the initial default ports are 8080 for HTTP, 8181 for HTTPS, and 4849 for the Administration Server.
  - The installation program will detect used ports and assign two others for you: Sun Java System Message Queue (by default, 7676), and IIOP (by default, 3700 for IIOP and 1060 and 1061 for IIOP/SSL). If these default port numbers are in use, the installation program will assign a random port number from the dynamic port range (note that this may not be the next available port number).

Starting previously-installed servers (UNIX) — unless you are replacing the previously installed server, you should start it before you begin the Sun Java System Application Server 8.2 installation process. This allows the installation program to detect ports that are in use and avoid assigning them for other uses.

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

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

For further compatibility information, see the *Sun Java System Application Server Enterprise Edition 8.2 Upgrade and Migration Guide*.

Bugs Fixed in the Enterprise Edition 8.2 Release

This section lists the customer-escalated issues resolved for the Sun Java System Application Server Enterprise Edition 8.2 product.

Bug Number	Description
6368745	AS: cannot upgrade from AS7 (Java ES 2) to AS8.2 (Java ES 5)
6432308	AS, JES5b7a, asupgrade from JES2 to JES5 fails
6378409	AS 8.2:backward compatibility broken because of jsf libraries that are included in 8.2
6371534	AS82EE:configure-ha-cluster hangs on Windows if the installation path contains space
6242761	The node agent cannot be started by init as documented without generating errors
6267772	Instructions about the configuration for Borland OptimizeIt are incorrect
6273226	Add text explaining to add the -Xrs jvm option to run a server/NA running as a windows s
6361145	Cannot upgrade LB plugin when in place upgrade from 8.1EE to 8.2EE
6362881	Installer doesn't give upgrade option when upgrade 8.1ur2 to 8.2ee
6325988	interop problem on first incoming RMI-IIOP request with FVD/codeBase
6363689	JES5 ASEE8.2 build03 - Unable to stop the instance

Bug Number	Description
6364900	Session value lost at the time of fail over when 1 web app includes a second web app
6370993	Session Failover collapses when the Application Context Root is Modified to "/" in Cluster
6373729	Appserver 8.1 code cannot communicate with WebLogic 9.0 because of ORB conflict
6377594	lookup problems with Weblogic initialcontext factory
6381538	Standalone Client fails with NPE
6406055	WARNING: "IOP00110205: (BAD_PARAM) Object reference came from foreign ORB" org.omg.CORBA.
6388329	JSP compilation error in Application Server after Access Manager upgrade
6419659	Requests not redirected correcly by the LB Plugin when transport-guarantee is CONFIDENTIAL
6390584	OutOfMemoryError: PermGen space
6401424	SEGV from service_plain_range in libns-httpd40.so when requested to byte serve a PDF file.
6401704	Need WebDAV Support for AppServer 8.#
6416478	jsp testsuite failue: javax.servlet.jsp.el.ELException
6438908	Header location corruption when relativeRedirectAllowed=true
6456553	java.lang.IllegalArgumentException when appending cookies to the response
6295010	Connections in the steady pool are not checked for idle timeout which conflicts with firewalls
6350435	Application Server fails to handle the failure of a database during an XA operation to two databases
6377830	setAutoCommit to false gets propagated when the same connection is being the next user
6399830	IT 319 : password alias feature not working in domain.xml
6360040	SJAS 8.x : AppServer LDAP Realm Bind User tends to access all groups and members
6370095	Cannot set acceptor-thread higher than 10.
6399365	InvokerServlet is not working in Enterprise Edition only
6303835	Excess logging: Misleading security messages in server log

Bug Number	Description
6349541	8.1 EE UR2 - SSL Listeners cannot be made to binding to a specific IP address...
6380040	Automated cleanup of logfiles required
6387278	Client authentication broken or not threadsafe(ProgrammaticLogin)
6407896	HttpServletRequestWrapper that overrides getUserPrincipal() causes ClassCastException
6321194	Round Robin policy not working
6362269	Verifier does not run correctly on Windows when installation path contains a space
6365888	Connections from the default connector connection pool are not enlisted in transactions
6369554	connection pool need to validate a connection before giving it to application
6370574	After AS upgrade with Confugre Later there is missing /var/opt/SUNWappserver directory
6371723	lbplugin leaks memory for all webserver version (more for Apache mod_loadbalancer)
6395390	Round Robin not happening on http requests that failover.
6402713	Loadbalancer failing to connect to HTTPS requests.
6409992	Upgrade failed with certificate from 8.1pe to 8.2EE
6413224	upgradetool skipped upgrade certificate option
6422893	HTTPS routing dosent work
6424051	Need to use existing admin credentials and MP in 8.xPE to 9.1 EE upgrade
6424053	8.XEE->9.1EE upgrade fails with a start-domain exception
6430394	Messages are lost when there is a n/w outage.
6444052	Integrate Generic RA for JMS version 1.5 into AS 8.2 EE
6444308	AS 8.1 UR2 EE-> 8.2 EE SS: Unable to start 8.2's domain1; wrong 8.1UR2's domain star
6444368	upgrade hangs from 8.0PE UR1 to 9.1 ee on win2003 side by side GUI
6446558	Manual transaction recovery does not work for connector-connection-pool resources.
6447895	Transction recovery not working for resources using embedded RA.
6454007	Change the input required for upgrade tool

Bug Number	Description
6455396	Node-agent and instances fail to startup after an 8.1EE->9.1EE SBS upgrade.
6374533	For performance and stability reasons Application Server should bundle XWSS 1.1 and not XWSS 1.0
6358422	Appserver 7.1/8.1 EE: web server LB proxy plug-in should properly support keep-alive connections
6382063	Memory leak in com.sun.enterprise.iiop.IORToSocketInfoImpl

Additional HADB Information

This section describes important additional information about the HADB implementation included in Application Server 8.2.

HADB Enhancements

- A new management command hadbm setadminpassword has been implemented to allow changing the password used for database administration. The command takes options indicating which management agent to use, and the old and new password. For more information, see the hadbm setadminpassword man page.

- The existing management command hadbm listpackages has been modified. Previously, the command took no operands, and listed all packages in the relevant management domain. The modifications introduces an optional package name operand, and lists only packages with that name. If the operand is not provided, all packages are listed. For more information, see the hadbm listpackages manpage.

- The existing management command hadbm createdomain has been modified. The *hostlist* operand is extended to also specify the port number of the management agent. In this way, the domain is completely specified using only the *hostlist* operand. The old behavior is still supported for backward compatibility. For more information, see the hadbm createdomain manpage.

- Some of the error messages from the management system have been modified. The modifications are intended to improve understandability, consistency and accuracy of the error messages. The actual modifications are not listed in these release notes.

- The installation and uninstallation behavior has been slightly changed. Installing or uninstalling the HADB should always preserve the softlink /opt/SUNWhadb/4, but this has not always been the case.

- The possibility of entering passwords on the command line as a command option is deprecated. This is relevant to all hadbm commands taking passwords as command line options. For hadbm commands, it has previously been possible to enter a password as:

  1. A password file
  2. A command line option
  3. An interactive input

  Method 2, the command line option, is considered unsafe, and is therefore deprecated. A warning message is issued if a password is entered in this way. Instead, use a method 1, password file, or method 3, interactive output. Using a password at the command line will become obsolete in the next release. Note this applies to all hadbm commands taking a command line password option.

- HADB has been upgraded to use JGroups Version 2.2, and its source code is distributed along with the HADB. To support online upgrade from a previous HADB version, both JGroups 2.1 and 2.2 are delivered with HADB. For JGroups 2.1, byte code is delivered only.

HADB File System Support

There are several important considerations if you want to configure HADB to use one of the following file systems:

- **ext2 and ext3**– HADB supports ext2 and ext3 file systems for Red Hat Application Server 3.0. For Red Hat Application Server 2.1, HADB supports only the ext2 file system.

- Veritas– When the Veritas File System is used on the Solaris platform, the message "WRN: Direct disk I/O mapping failed" is written to the history files. This message indicates that HADB cannot turn on direct I/O for the data and log devices. Direct I/O is a performance enhancement that reduces the CPU cost of writing disk pages. It also causes less overhead of administering dirty data pages in the operating system.

  To use direct I/O with the Veritas File System, use one of the following:

  - Create the data and log devices on a file system that is mounted with the option mincache=direct. This option applies to all files created on the file system. See the mount_vxfs(1M) command for details.

  - Use the Veritas Quick I/O facility to perform raw I/O to file system files. See the *VERITAS File System 4.0 Administrator's Guide for Solaris* for details.

    Note that these configurations have not been tested with Application Server 8.2.

    Refer to the *Application Server Enterprise Edition High Availability Administration Guide* for information about installing and configuring HADB with Application Server software.

Upgrading the High Availability Database

Sun Java System Application Server Enterprise Edition 8.2 Release Notes • November 2009

▼ Pre-upgrade Tasks/Data Migration

Before You Begin   Users should keep the HADB history files, management agent configuration files, log files and repository, and all the data devices outside the installation path. If not, this should be done prior to the upgrade. To move the management repository and configuration files:

1   **Stop all the old management agents and keep the HADB nodes running.**

2   **On each host, move the repository directory to the new location.**

3   **On each host, copy the** dbconfig **directory to the new location.**

4   **On each host, update the** mgt.cfg **file, and set the correct path for** dbconfig **and repository directory.**

5   **Start the management agents using the updated** mgt.cfg **file.**

▼ Upgrade Procedure

To upgrade from HADB version 4.4.x to version 4.4.3, perform the following steps:

1   **Perform the pre-upgrade tasks mentioned above as necessary.**

2   **Install HADB version 4.4.3 on all HADB hosts (on another path than that of version 4.4.x, for instance on** /opt/SUNWhadb/4.4.3**).**

3   **Install the HADB 4.4.3 version on the** hadbm **client hosts, if they are different than that of the HADB hosts.**

4   **Stop all management agents running on all HADB hosts.**

5   **Start the management agent processes using the new version's software, but with the old configuration files. In the remaining steps, please use the** hadbm **command found in the new version's** bin **directory.**

6   **Register the package in the management domain (default package name becomes V4.4, so another package name may be required to avoid conflicts with existing packages having the same name):**

```
hadbm registerpackage --packagepath=/opt/SUNWhadb/4.4.3 V4.4.3
```

7   **Run the** hadbm listpackages **command and check that the new package is registered in the domain.**

8 **Restart the database with the new** hadbm **version 4.4.3. If it is necessary to move the devices and history files, run online upgrade combined with setting new paths for devices and history files in one single operation:**

```
hadbm set packagename=V4.4.3,devicepath=new_devpath,
historypath=new_histpath
```

Otherwise, if the devices and history files are already outside of the installation directory, run the following command, which only does a rolling restart of the nodes:

```
hadbm set packagename=V4.4.3 database name
```

9 **Check that the database status is "running" (using the** hadbm status **command) and that it functions normally, serving the client transactions.**

10 **If everything is working, the old installation can be removed later. Before unregistering the old package, remove all references to the old package from the** ma **repository. Otherwise,** hadbm unregisterpackage **will fail with "package in use." A dummy reconfiguration operation, for instance,** hadbm set connectiontrace=*same as previous value* **will remove all references to the old package. Now, unregister the old package:**

```
hadbm unregisterpackage [--hosts=host-list] old pacakge name
```

11 **Remove the old installation from the file system.**

▼ Testing the Upgrade

On Solaris, to test that the upgrade was successful, check that the upgrade was performed properly:

1 **Ensure that the running processes use the new binaries. Check the following in all HADB nodes:**

```
new path/bin/ma -v
new path/bin/hadbm -v
```

2 **Check whether the database is running. The following command should show that all the HADB nodes are in a "running" state.**

```
new path/bin/hadbm status -n
```

3 **Ensure that the products using HADB have changed their pointers to point to the new HADB path.**

4 The products using the HADB can run their upgrade tests to verify the HADB upgrade is also working.

After an online upgrade, if the new version does not work properly, go back to using the previous HADB version. However, if there has been a change to the management agent repository, the HADB itself can be downgraded, but the new management agent must be kept running.

Special Deployment and Upgrade Information

This section lists additional information about HADB deployment and upgrading.

-
-

Deployment

- Store device, log and history files on local disks only, do not use remote-mounted file systems.

- If more than one node is placed on a host, it is recommended to keep the devices belonging to each node on different disks. Otherwise, the disk contention would reduce the performance. Symptoms of this problem can be seen in the history files by the messages such as BEWARE - last flush/fputs took too long. When one single node has more than one data device file, it is recommended to use separate disks for these device files.

- Use local disks (preferably separate disk than the one used for data devices) to install HADB binaries on HADB hosts. NFS delays or disk contention may cause node restarts with warning, "Process blocked for nnn, max block time is nnn" in the history files.

- Do not place the HADB devices, history files, management agent directories and agent configuration files in the HADB package path. This will cause problems when upgrading to newer versions and deleting the old package path.

- This release of HADB is officially supported for a maximum of 28 nodes; 24 active data nodes with 4 spares.

- We recommend using the same version for the JDBC driver and the HADB server.

- We do not support IPv6, only IPv4.

- The command line length on Windows is restricted to 2048 bytes.

- The network must be configured for UDP multicast.

- Due to excessive swapping observed in RedHat Enterprise Linux 3.0, updates 1 through 3, we do not recommend it as a deployment platform. The problem is fixed in RedHat Enterprise Linux 3.0 update 4.

- Possibility of running NSUP with real time priority.

  The node supervisor (NSUP) processes (clu_nsup_srv) ensure the high availability of the HADB with the help of exchanging "heartbeat" messages in a timely manner. The timing gets affected when an NSUP is colocated with other processes causing resource starvation.

The consequence is false network partitioning and node restarts (preceded by a warning "Process blocked for n seconds" in history files) resulting in aborted transactions and other exceptions.

To solve this problem, `clu_nsup_srv` (found in `installpath/lib/server`) must have the suid bit set and the file must be owned by root. This is achieved manually by the commands:

```
# chown root clu_nsup_srv
# chmod u+s clu_nsup_srv
```

This causes the `clu_nsup_srv` process to run as the user root when started, and this in turn allows the process to automatically give itself real-time priority after startup. To avoid any security impact by using setuid, the real-time priority is set in the very beginning and the process falls back to the effective uid once the priority has been changed. Other HADB processes will lower their priority to timeshare priority.

If NSUP could not set the real-time priority, it issues a warning, "Could not set realtime priority" (unix: errno will be set to EPERM), which is written out in `ma.log` file and continues without real-time priority.

There are cases where it is not possible to set real-time priorities; for example:

- When installed in Solaris 10 non-global zones
- When `PRIV_PROC_LOCK_MEMORY` (Allow a process to lock pages in physical memory) and/or `PRIV_PROC_PRIOCNTL` privileges are revoked in Solaris 10
- Users turn off setuid permission
- Users install the software as tar files (nonroot install option for the `App.server`)

The `clu_nsup_srv` process is not CPU consuming, its footprint is small and running it with real-time priority will not impact performance.

- Configuring IP network multipathing for HADB for Solaris (tested on Solaris 9 only).

Sun recommends that Solaris hosts running HADB be set up with network multipathing in order to ensure the highest possible network availability. Network multipathing setup is covered in detail in the *IP Network Multipathing Administration Guide*. If you decide to use multipathing with HADB, refer to the Administering Network Multipathing section of the *IP Network Multipathing Administration Guide* in order to set up multipathing before you proceed with adapting the multipathing setup for HADB as described below. The *IP Network Multipathing Administration Guide* is part of the Solaris 9 System Administrator Collection, and can be downloaded from http://docs.sun.com.

- **Set network interface failure detection time**

    For HADB to properly support multipathing failover, the network interface failure detection time must not exceed 1000 milliseconds as specified by the `FAILURE_DETECTION_TIME` parameter in /etc/default/mpathd. Edit the file and change the value of this parameter to 1000 if the original value is higher:

    ```
    FAILURE_DETECTION_TIME=1000
    ```

In order for the change to take effect, issue the following command:

```
pkill -HUP in.mpathd
```

- **IP addresses to use with HADB**

  As described in the *Solaris IP Network Multipathing Administration Guide*, multipathing involves grouping physical network interfaces into multipath interface groups. Each physical interface in such a group has two IP addresses associated with it: a physical interface address and a test address. Only the physical interface address can be used for transmitting data, while the test address is for Solaris internal use only. When `hadbm create --hosts` is run, each host should be specified with only one physical interface address from the multipath group.

- **Example**

  Assume that Host 1 and Host 2 have two physical network interfaces each. On each host, these two interfaces are set up as a multipath group, and running `ifconfig -a` yields the following:

  Host 1

  ```
  bge0: flags=1000843<mtu 1500 index 5 inet 129.159.115.10 netmask ffffff00
  broadcast 129.159.115.255 groupname mp0
  bge0:1: flags=9040843<mtu 1500 index 5 inet 129.159.115.11 netmask ffffff00
  broadcast 129.159.115.255
  bge1: flags=1000843<mtu 1500 index 6 inet 129.159.115.12 netmask ffffff00
  broadcast 129.159.115.255 groupname mp0
  bge1:1: flags=9040843<mtu 1500 index 6 inet 129.159.115.13 netmask ff000000
  broadcast 129.159.115.255
  ```

  Host 2

  ```
  bge0: flags=1000843<mtu 1500 index 3 inet 129.159.115.20 netmask ffffff00
  broadcast 129.159.115.255 groupname mp0
  bge0:1: flags=9040843<mtu 1500 index 3 inet 129.159.115.21 netmask ff000000
  broadcast 129.159.115.255
  bge1: flags=1000843<mtu 1500 index 4 inet 129.159.115.22 netmask ffffff00
  broadcast 129.159.115.255 groupname mp0
  bge1:1: flags=9040843<mtu 1500 index 4 inet 129.159.115.23 netmask ff000000
  broadcast 129.159.115.255
  ```

  Here, the physical network interfaces on both hosts are the ones listed as `bge0` and `bge1`. The ones listed as `bge0:1` and `bge1:1` are multipath test interfaces (they are thus marked as DEPRECATED in the `ifconfig` output), as described in the *IP Network Multipathing Administration Guide*.

  To set up HADB in this environment, select one physical interface address from each host. In this example. we choose `129.159.115.10` from host 1 and `129.159.115.20` from host 2. To create a database with one database node per host, use the following argument to `hadbm create`:

```
--host 129.159.115.10,129.159.115.20
```

To create a database with two database nodes on each host, use the following argument:

```
--host 129.159.115.10,129.159.115.20,129.159.115.10,129.159.115.20
```

In both cases, the `ma.server.mainternal.interfaces` variable on both hosts should be set to `129.159.115.0/24`.

Online Upgrade from 4.4.1 to 4.4.2

It is not possible to upgrade from 4.2 or 4.3 to 4.4 online. However, 4.4 supports online upgrade for the future versions. To upgrade from 4.4.1 to 4.4.2, perform the following steps:

1. Install 4.4.2 on all HADB hosts (On another path than that of 4.4.1 – for instance `/opt/SUNWhadb/4.4.2-6`).

2. Install the new version on the `hadbm client` hosts.

3. Stop all management agents running on the HADB hosts.

4. Start the management agent processes using the new version's software, but with the old configuration files. In the remaining steps, please use the `hadbm` command found in the new version's `bin` directory.

5. Register the package in the management domain (default package name here becomes `V4.4`, so another package name may be required to avoid conflicts with existing packages having the same name):

   ```
   hadbm registerpackage --packagepath=/opt/SUNWhadb/4.4.2-6 V4.4.2
   ```

6. Restart the database with the new version (the following command does a rolling restart of the nodes):

   ```
   hadbm set packagename=V4.4.2 database_name
   ```

7. Check that the database status is "running" (using the command `hadbm status`) and that it functions normally, serving the client transactions.

8. If everything works, the old installation can be removed later.

   Before unregistering the old package, remove all references to the old package from the `ma` repository. Otherwise, `hadbm unregisterpackage` will fail with "package in use." A dummy reconfiguration operation, for instance, hadbm set connectiontrace=<same_as_previous_value> will remove all references to the old package. Now, unregister the old package:

   ```
   hadbm unregisterpackage [--hosts=<host_list>] <old_package_name>
   ```

   Remove the old installation from the file system, as described in the HADB installation instructions (http://clustra.norway.sun.com/intraweb/download/products/hadb/packages/pdf/4.4.2-6.pdf).

Known SQL Limitations

- It is not possible to create a UNIQUE secondary index on a table.

- The expression (DISTINCT column) is not allowed in an aggregate expression, unless this is the only selected expression.

- All tables must be created with a primary key specification (that is, tables without primary keys are not supported).

- FULL OUTER JOIN is not supported.

- IN subqueries that are table subqueries are not supported; for example:

  ```
  SELECT SNAME FROM S WHERE (S1#,S2#) IN (SELECT S1#,S2# FROM SP
  WHERE P#='P2')
  ```

- Constraints other than NOT NULL and PRIMARY KEY are not supported.

- It is possible to assign a new owner to a resource. When doing this, however, privileges granted to the current owner are not granted to the new owner.

- Two or more nested NOT EXISTS subqueries where each subquery is not (directly) correlated to outer level of queries, is not supported.

- Column privileges are not supported.

- Row value constructors are allowed only in a VALUES clause.

- Subqueries are not accepted as value expressions in row value constructors.

- The following data types cannot be used when creating primary keys:

  - REAL
  - FLOAT
  - DOUBLE PRECISION
  - DECIMAL
  - NUMERIC

High Availability Load Balancing

The Application Server includes load balancing for HTTP, IIOP, and JMS clients; HTTP session failover support; EJB clustering and failover support; highly available EJB timers; distributed transaction recovery; support for rolling application upgrades; and a high availability database for storing the transient state of J2EE applications.

Availability allows for failover protection of Application Server instances in a cluster. If one Application Server instance goes down, another Application Server instance takes over the sessions that were assigned to the unavailable server. Session information is stored in the HADB. HADB supports the persistence of HTTP sessions, Stateful Session Beans, and Single Sign On credentials.

Compatibility Issues

In the next major release of the Sun Java System Application Server Enterprise Edition the following incompatibilities will be introduced:

- While the HTTP Service will continue using a DNS cache for better performance, monitoring of the DNS cache will not be available.
- The support for HTTP file caching will be revamped, resulting in changes to configuration and monitoring.
- The format for the access log rotation suffix will be changed to the format supported by date and time objects as specified in `http://java.sun.com/j2se/1.5.0/docs/api/java/text/SimpleDateFormat.html`. The default value in this release, "%YYYY;%MM;%DD;-%hh;h%mm;m%ss;s," will continue to be supported but no other variations will be supported.
- Any `domain.xml` elements, attributes and properties no longer supported will be flagged as warnings in the server log and in the upgrade log file as having been deprecated.
- The `server.http-service.dns` node will no longer be available in the monitoring view.
- Some of the attributes from the `server.http-service.file-cache` node may be removed. Consequently, any `asadmin` monitoring command trying to access removed attributes from these nodes will fail.

Deploytool

Deploytool will no longer be available. The equivalent functionality is available in the NetBeans IDE. For more information, please see J2EE 1.4 tutorial for NetBeans 4.1 at `http://www.netbeans.org/kb/`.

Verifier

- Verifier GUI mode (invoked by `verifier -u`) will no longer be available. The equivalent functionality will be available in the NetBeans IDE.
- The default mode for application verification when using verifier tool will change from "Verify J2EE rules" to "Verify J2EE rules and Sun Application Server Configuration Rules." In other words, by default verifier will test whether an application meets J2EE rules and is configured to run on Sun Application Server. The verifier command will have a command-line switch to test an application for J2EE rules only.

Classloader Changes

In the current release, the JAR and directory entries added to `classpath-prefix`, `server-classpath`, and `classpath-suffix` attributes of `domain.xml` (application server configuration file) are available in the JVM system classpath. An application depending on this behavior might be using the following methods from the class `java.lang.ClassLoader` to access classes or other resources from JVM system classpath:

- `getSystemClassLoader()`

- `getSystemResource()`
- `getSystemResourceAsStream()`
- `getSystemResources`

In the next major release, the JAR and directory entries added to `classpath-prefix`, `server-classpath`, and `classpath-suffix` will no longer be available in the JVM system classpath. If an application uses one of the methods mentioned above, Sun strongly recommends using an equivalent method that does not assume that the resources are available in the system classpath. The equivalent methods that do not rely on the JVM system classpath are available in `java.lang.ClassLoader` and should be used when possible; for example:

EXAMPLE 2–1 Old Code

```
java.net.URL url = ClassLoader.getSystemResource
("com/acme/tools/tools.properties");
```

EXAMPLE 2–2 Suggested Change

```
java.net.URL url = this.getClass().getClassLoader().getResource
("com/acme/tools/tools.properties");
```

If it is not possible to change the code, then you may choose to use a new configuration option that will be added in the next release to set JVM system classpath.

Web Service Security Configuration

Security for Web services can be configured using the files `wss-client-config.xml` and `wss-server-config.xml`. Please note that the content and names of these configuration files are unstable and likely to change. The equivalent functionality will continue to be available.

J2EE Support

The Sun Java System Application Server Enterprise Edition 8.2 supports the J2EE 1.4 platform. The following table describes the enhanced APIs available on the J2EE 1.4 platform.

TABLE 2–5 APIs Available on the J2EE 1.4 Platform

API	Description
Components	
Application and Application Client	Implementation of standard deployment descriptors by means of XML schemas
Enterprise JavaBeans (EJB) 2.1	Timer service and EJB Web-service endpoint
Java Servlet 2.4	Web-service endpoint filter

TABLE 2–5 APIs Available on the J2EE 1.4 Platform   *(Continued)*

JavaServer Pages (JSP) 2.0 architecture	Expression language and tag library
J2EE Connector Architecture 1.5	Inbound resource adaptor and Java Message Service (JMS) pluggability
Web Services	
Java Web Services Developer Pack 1.5	Integrated toolkit for building, testing and deploying XML applications, Web services, and Web applications
Java API for XML-based Remote Procedure Calls (JAX-RPC) 1.1	Mapping for WSDL and Java technology and support for development of Web-service clients and endpoints
WS-I Basic Profile 1.0	The enabling element for interoperability using WSDL and SOAP
SOAP with attachment API for Java (SAAJ) 1.2	An API for SOAP-based messaging; fosters the creation of SOAP messages with attachments
Java APIs for XML Registries (JAXR) 1.0	A uniform and standard API for accessing XML registries, such as those for Universal Description Discovery and Integration (UDDI and ebXML)
Other	
J2EE Deployment 1.1	Standard APIs that enable deployments of J2EE components and applications
J2EE Management 1.0	Definitions for the information model for managing the J2EE platform
Java Management Extensions (JMX) 1.2	Standard management API
Java Authorization Contract for Containers (JACC) 1.0	Definitions of security contracts between a J2EE Application Server and the authorization policy provider
Java API for XML Processing (JAXP) 1.2	An API with which applications can parse and transform XML documents; also adds support for processing of XML schemas
JMS 1.1	A messaging standard that enables J2EE application components to create, send, receive, and read messages; also adds support for uniform APIs for queues and topics
JavaMail 1.3	A set of abstract classes that model a mail system; also includes minor updates to the APIs

Switching to Another Supported Java Version

Sun Java System Application Server 8.2 requires J2SE 5.0 or greater as the underlying JVM. If you want to switch from one Java version to another, perform the following general steps. (Windows and Unix)

▼ To switch to another supported Java version

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

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

2   Completely stop the Application Server.

You can use the following command line:

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

Alternatively, you can use the Administration Console GUI:

a.   Click the Application Server node.

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

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

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

5   Restart the Application Server.

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

High Performance

The Application Server includes a high performance EJB container, Web container and services, and supports concurrent message delivery with the Sun Java System Message Queue software.

Scalability

The Application Server supports horizontal scalability through clustering of server instances and request load balancing. It also achieves class leading vertical scalability supporting large multi-processor machines. The integrated message broker can be clustered for better scalability and availability. Client access from HTTP clients, RMI/IIOP based Rich Client Applications, Web Services Clients, and JRM Clients can be load balanced to Application Server clusters.

JavaServer Faces 1.1 Support

The Sun Java System Application Server Enterprise Edition 8.2 supports JavaServer Faces 1.1 technology. The JavaServer Faces technology consists of a set of server-side APIs that represent user-interface components that manage their state, event, handling, and input validation. The APIs also define page navigation and support internationalization and accessibility. You can add custom user-interface components with a JSP custom tag library.

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

3

Known Issues and Limitations

This chapter describes known problems and associated workarounds for the Sun Java System Application Server Enterprise Edition 8.2 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

This section describes known administration issues and associated solutions.

The `package-appclient` **script does not work if** `domain1` **is not present. (ID 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 by asenv.conf. If domain1 is deleted and a new domain created, the AS_ACC_CONFIG variable is not updated with the new domain name, which causes the package-appclient script to fail.

Solution

Do one of the following:

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

Installing the Load Balancing Plugin will overwrite an existing plugin. (ID 6172977)

Description

If you install the Load Balancing plugin against an installation of the Application Server that already has a Load Balancer plugin installed (for example, from 7.1EE), then the 8.2EE plugin will silently replace any existing Load Balancer, even if you have created a new server instance in which to run the plugin.

The plugin files are installed by default under the install_dir/plugins/lbplugin directory, which means that only one version of a plugin can be used with any one Application Server installation. Note that the console installer does display a message indicating that an uninstall is being performed, but this message can sometimes be easy to miss.

Solution

Not everyone will encounter this problem. If you do encounter the problem, remove the old Application Server installation and do a fresh install rather than doing an upgrade installation.

Several Changes in `asadmin` **script in JES3 Application Server 8.2 compared to JES2 AS7 (ID 6189433, 6189436)**

There have been several changes made to the asadmin command in Application Server 8.2 compared to Application Server 7.x. For example, in 7.x, the command to start a server instance is:

```
asadmin start-instance
```

In 8.2, the equivalent command is:

```
asadmin start-domain --user admin domain1
```

Refer to the following documents for complete information about the latest asadmin command syntax:

- *Sun Java System Application Server Enterprise Edition 8.2 Administration Guide*
- *Sun Java System Application Server Enterprise Edition 8.2 Reference Manual*
- *Sun Java System Application Server Enterprise Edition 8.2 Upgrade and Migration Guide*

Default ports changed in Application Server (ID 6198555)

Description

When upgrading to JES5/Application Server 8.2 from JES2/Application Server 7.x, you may experience incompatibilities or errors because the default ports have changed.

Solution

Refer to "Other Requirements" on page 19 earier in these notes for a listing of the default ports used in Application Server 8.2.

Cannot restore backed-up domain with another name. (ID 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 8.2 work only for backing up and restoring the same domain on the same machine.

Starting Application Server with additional JMX Agent is not supported. (ID 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 VM. An undesirable side-effect of this is that the administration functions are affected adversely, and the Application Server administration GUI and CLI may produce unexpected results. The problem is that there are some conflicts between the built in jmx-connector server and the new jmx-connector server.

Solution

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

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

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

For more information, refer to the *Sun Java System Application Server 8.2 Administration Guide*.

On UNIX, overly restrictive execute permissions on Application Server start and stop scripts. (ID 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. (ID 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.

Java Home Setting inside Configuration does not take effect. (ID 6240672)

Description

Application Server domains/servers do not use the JDK pointed to by `java-home` attribute of `java-config` element of associated configuration.

Solution

The JDK used by the Application Server processes for all the domains in a given server installation is determined by the `appserver-installation-dir/config/asenv.conf` file. The property `AS_JAVA` in this file determines the JDK used and is set at the time of installation. If a different JDK is to be used by Application Server processes after the installation is completed, this value can be modified to point to another JDK. Note that all domains in this installation will be affected by this change.

Note – Manual changes to `asenv.conf` file are not checked for validity and hence care should be exercised while changing them. Check the product documentation for minimum JDK version requirements when modifying the value for `AS_JAVA`.

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

Description

This problem is caused by a wrong value for `%CONFIG_HOME%`.

Solution

1. Rename the existing 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.

The .asadmintruststore file is not described in the Application Server documentation. (ID 6315957)

Description

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.

Domain fails to start when create-domain master password has special characters. (ID 6345947)

Description

Domain does not start when the domain's master password contains the percent (%) character.

Solution

The domain's master password should not contain a percent character (%). This applies when creating a new domain or changing the master password for an existing domain.

Load Balancer configuration changes in magnus.conf and obj.conf get overwritten (ID 6394181)

Description

After creating a secure http-listener and installing lbplugin, the magnus.conf and obj.conf files under the webserver_instance_dir/config are getting modifiedand the lbplugin contents are getting removed.

The installer modifies the magnus.conf and obj.conf configuration files on the Application Server as part of the installation of the load balancer plugin. If you log in to the Application Server admin console and try to manage the instance configuration for the instance on which the load balancer has been installed, the Application Server gives a warning message stating that it has detected a manual edit in the configuration. This warning is in fact referring to the changes made by the installer.

Solution

Verify that the changes made by the installer have not been overwritten.

Unable to start server instance on Txxxx series (Niagara) machines. (6860787)

Description

On T2000 and T5xxx machines there is possibility of JMX timeouts happening and preventing synchronization. The process of binding the JMX/JMX connection to the domain administration server (DAS) is bounded to 5 seconds. If the JMX connection plus the handshake takes more that 5 seconds, the JMX connection might be deemed a failure. This seems to happen more on a T-series machine, especially if the ORB+SSL initialization takes more time. However, this type of failure also happens in other machine/architecture situations.

Solution

1. Increase the default values from 5 seconds. 10 seconds should be acceptable, unless the target DAS/network is extremely loaded. (2-4 seconds is the approximate subsystem setup overhead for the ORB/SSL/RMI connection even on a normal Sun v240.)

2. In Application Server, use the flags (as JVM options) to set the JMX timeout of the synchronization, and other JMX connections if needed.

   - Set the timeout by using DJMXCONNECTOR_TIMEOUT_MILLISEC=<timeout>

   - Turn the timeout off and revert to the old behavior by using DDISABLE_JMXCON_THREAD=true

. As these are JDK options, the synchronization can happen as follows:

- As a node agent starting a Java program using INSTANCE-SYNC-JVM-OPTIONS as the synchronization JVM flag

  See documentation for details on the option. You can set this property with either of the above values. However due to existing bug 6857893, INSTANCE-SYNC-JVM-OPTIONS can currently only take one JVM value

- As a normal JDK option to the instance/cluster configuration

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. (ID 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. (ID 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).

Must start Apache Web Server as root. (ID 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.

Addition to instructions for using `openssl` **with Apache Web Server 2.0 on Solaris. (ID 6308043)**

After installing Apache 2.0 and the load balancer plug-in, edit `ssl.conf` and `sll-std.conf` as follows:

Replace the line:

```
<VirtualHost _default_:9191>
```

with

```
<VirtualHost machine_name:9191>
```

Where *machine_name* is the name of your machine and 9191 is a security port number.

Application Client

This section describes known application client issues and associated solutions.

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

Dynamic content technology such as CGI-bin and SHTML functionality not supported. (ID 6373043)

Description

Dynamic content technologies, such as CGI-bin and SHTML, are no longer supported.

Solution

Use JSP and Web service technologies instead.

Bundled Sun JDBC Drivers

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

Applications using the TRANSACTION_SERIALIZABLE isolation level with the bundled Sun driver for Microsoft SQL Server may hang when using a prepared statement to update if two parallel transactions are running and one of them is rolled back. (ID 6165970)

To set a desired isolation level for a connection, the corresponding connection pool must be created at that same isolation level. See the *Sun Java System Application Server Enterprise Edition 8.2 Administration Guide* for details about configuring connection pools.

PreparedStatement Errors. (ID 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 Enterprise Edition 8.2 Administration Guide* for details about configuring connection pools.

Description 2

Related to the PrepardStatement 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 Enterprise Edition 8.2 Administration Guide* for instructions.

Problems setting isolation level with the bundled Sun driver for Sybase Adaptive Server. (ID 6189199)

Description

- Applications using the TRANSACTION_SERIALIZABLE isolation level with the bundled Sun driver for Sybase Adaptive Server may hang when using a prepared statement to update if two parallel transactions are running and one of them is rolled back. Connection rollback fails with following message, and the rolled back connections cannot be used anymore:

  ```
  java.sql.SQLException: [sunm][Sybase JDBC Driver]Request cannot be submitted
  due to wire contention
  ```

- Sybase Adaptive Server does not support the TRANSACTION_REPEATABLE_READ isolation level. However, querying DatabaseMetaData, the bundled Sun driver returns that this isolation level is supported by the database. Applications using the this isolation level will fail.

- Applications using the bundled Sun driver cannot set the TRANSACTION_READ_UNCOMMITTED isolation level. The application throws the following exception on the first DataBaseMetaData access:

  java.sql.SQLException: [sunm][Sybase JDBC Driver][Sybase]The optimizer could not find a unique index which it could use to perform an isolation level 0 scan on table 'sybsystemprocs.dbo.spt_server_info'.

Solution

None at this time.

On Solaris 10 and Enterprise Linux 3.0, the Sun bundled Oracle JDBC driver does not allow the creation of a connection. (ID 6247468)

Set the following property on the JDBC connection pool when using the SUN JDBC oracle datasource (com.sun.sql.jdbcx.oracle.OracleDataSource):

<property name="serverType" value="dedicated"/>

The value of the property depends upon the way the Oracle server's listener is configured. If it is configured in the "shared" mode, the above value needs to change to "dedicated".

java.lang.SecurityException**: Sealing violation exception (ID 6554602)**

Description

Starting with JDBC 10.2 drivers, having more than one JDBC jar file in the CLASSPATH may result in a java.lang.SecurityException: Sealing violation exception.

Detailed explanation from Oracle is documented in the following Oracle Document ID:

Note:405446.1
Subject: JDBC Driver 10.2 Uses Sealed JAR files and May Cause SecurityException Sealing Violation

Solution

(Suggested by Oracle) Make sure that the CLASSPATH includes only one JDBC driver JAR file.

Connectors

This section describes known J2EE connector architecture issues and associated solutions.

After restarting a DAS instance, undeploying the connector module fails when cascade is set to false. (ID 6188343)

Description

In this scenario, a standalone or embedded connector module is deployed in DAS and connector connection pools, and resources are created for the deployed module. After restarting the DAS instance, undeploying the connector module fails when cascade is set to false with the following exception:

```
[#|2004-10-31T19:52:23.049-0800|INFO|sun-appserver-ee8.1|javax.enterprise.
system .core|_ThreadID=14;|CORE5023: Error while unloading application [foo]|#]
```

Solution

Use cascaded undeploy (set the cascade option to true) for undeploying standalone and embedded connectors after restart of the DAS instance.

JMS `create-jms-resource`; CLI doesn't set the default values correctly (ID 6294018)

Description

Because you cannot specify the minimum pool size and maximum pool size when creating a new JMS resource from the command line with the asadmin `create-jms-resource` command, the asadmin command is supposed to create the resource using the default pool size values (minimum 8, maximum 32). However, this is not the case. Instead, creating the resource from the command line results in default minimum and maximum pool sizes of 1 and 250, respectively.

Solution

After creating a JMS resource from the command line, use the admin console to modify the minimum and maximum pool size values.

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 `AltJDBCConnectionPoolStats`. 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`. (ID 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.

Logging options documentation incorrect (ID 6463965)

The *Sun Java System Application Server Enterprise Edition 8.2 Performance Tuning Guide* incorrectly states the following about Log Options:

> The Administration GUI provides the following two logging options:
>
> - *Option 1* – Log stdout (`System.out.print`) content to the event log
> - *Option 2* – Log stderr (`System.err.print`) content to the event log

These log options no longer exist in Application Server Enterprise Edition 8.2.

Conflicting information regarding HTTP file cache feature in Application Server 8.2 (ID 6474799)

The Application Server Enterprise Edition 8.2 documentation discusses an HTTP file caching feature, in "HTTP File Cache" in *Sun Java System Application Server Enterprise Edition 8.2 Performance Tuning Guide*. However, this feature was not included in Application Server Enterprise Edition 8.2. Note that this feature has been reintroduced in Application Server 9.0.

Documentation on getting a physical Connection from a wrapped Connection is no longer correct (ID 6486123)

As a result of other defects (possibly 6295215) the code provided in the "Obtaining a Physical Connection from a Wrapped Connection" in *Sun Java System Application Server Enterprise Edition 8.2 Developer's Guide* section of Chapter 11, "Using the JDBC API for Database Access," in *Sun Java System Application Server Enterprise Edition 8.2 Developer's Guide* is not correct. Specifically, the line:

```
Connection drivercon = ds.getConnection(con);
```

should now read:

```
Connection drivercon = ((com.sun.gjc.spi.DataSource)ds).getConnection(con);
```

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)

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

Description

```
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.

Shared memory segments locked and cannot be paged out. (ID 5052548)

Description

HADB 4.3-0.16 and later is configured to use Intimate Shared Memory when it creates and attaches to its shared memory segments (uses the SHM_SHARE_MMU flag). The use of this flag essentially locks the shared memory segments into physical memory and prevents them from being paged out. This can easily cause problems with installations on low end machines.

Therefore if a developer has a machine with 512MB of memory and plenty of swap space available when using Application Server7.0 EE, and then installed 7.1 EE or later, he or she will encounter problems configuring the default `clsetup` cluster (which creates two HADB nodes, each with a `devicesize` of `512`, which results in there not being enough physical RAM to support the shared memory that both nodes require.

Solution

Make sure you have the recommended amount of memory when co-locating Application Server and HADB. See "HADB Requirements and Supported Platforms" on page 18 for more information.

`hadbm set` does not check resource availability (disk and memory space). (ID 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. (ID 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. (IDs 6173886, 6253132)

Description

If running the management agent on a host with multiple network interfaces, the `create domain` 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.

Directories need to be cleaned up after deleting an HADB instance. (ID 6190878)

Description

After deleting an HADB instance, subsequent attempts to create new instances with the `configure-ha-cluster` command fail. The problem is that old directories are left from the original HADB instance in `ha_install_dir/rep/*` and `ha_install_dir/config/hadb/instance_name`.

Solution

Be sure to manually delete these directories after deleting an HADB instance.

Starting, stopping, and reconfiguring HADB may fail or hang. (IDs 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 (`nomandevice`) 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"**. (ID 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. (ID 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. (ID 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. (ID 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. (ID 6271063)**

Description

Regarding install/removal of HADB c package (Solaris: `SUNWhadbc`, 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. (ID 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. (ID 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

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. (ID 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. (ID 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. (ID 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`) from the Microsoft support pages.

When running `hadbm start <db_name>`**, part of the inputted password is displayed without
being masked. (ID 6303581, 6346059, 6307497)**

Description

It is possible when a machine is under load that the masking mechanism fails and some
characters from the password being entered are exposed. This poses a minor security risk, and
the password should always be masked.

Solution

Put the passwords in their own password files (the method normally recommended since
Application Server 8.1) and refer to these with either the `--adminpassword` or
`--dbpasswordfile` options.

JES5 HADB installed in Global Zone not accessible from Sparse Local Zones (ID 6460979)

Description

When the Application Server is installed in a Solaris Global Zone to `/usr/SUNWappserver`, the
HADB component installed with that Application Server instance will not be available in Sparse
Local Zones.

The problem is that HADB is installed to `/opt/SUNWhadb` in the Global Zone, but this directory
is not readable from Sparse Local Zones. Unfortunately, the HADB bundle in JES5 is not
relocateable.

Solution

Because the Application Server HADB component is not relocatable, the HADB component
must be installed separately in each Sparse Local Zone from which you want to access HADB.

Installation

This section describes known installation issues and associated solutions.

Installation shutdown hanging on some Linux systems after clicking the "Finish" button. (ID 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. (ID 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
   ```

Application Server cannot be set up on RHLAS 3.0 and RHLAS 4.0 without `compat-libstdc++`**. (ID 6396102)**

Description

Installing Application Server Enterprise Edition 8.2 on a Red Hat Linux Advanced Server (RHLAS) 3.0 or 4.0 system will fail if the `compat-libstdc++` library is not already installed on the system. Application Server requires t he `compat-libstdc++` library on RHLAS systems, but this library is not installed by default. Note that this is only a problem on RHLAS systems.

Solution

Download and install the `compat-libstdc++` RPM from http://rpm.pbone.net/index.php3/stat/4/idpl/843376/com/compat-libstdc++-7.3-2.96.118.i386.rpm.html before installing Application Server software.

`lbplugin (`libpassthrough.so`) can not be used when server is running in 64-bit mode (ID 6480952)`

Description

When running Application Server Enterprise Edition 8.2 with Web Server 7.0 in 64–bit mode, attempts to run a 64–bit version of the load balancer plugin fail with the following error:

```
failure: CORE2253: Error running Init function load-modules: dlopen
of /export/home/mareks/opt/webserver7/plugins/lbplugin/bin/libpassthrough.so
failed (ld.so.1: webservd: fatal: /export/home/mareks/opt/webserver7/plugins/
lbplugin/bin/libpassthrough.so: wrong ELF class: ELFCLASS32)
failure: server initialization failed
```

The problem is that there is no 64–bit load balancer plugin for Application Server Enterprise Edition 8.2, and the 64–bit Web Server requires 64–bit plugins.

You can determine whether Web Server is running in 64–bit or 32–bit mode by using the following command:

```
wadm get-config-prop --user=admin --config=xxx --password-file=xxx platform
```

Solution

No 64–bit load balancer is planned for Application Server Enterprise Edition 8.2. To work around the problem, either use the Web Server 7.0 reverse proxy feature, or configure Web Server 7.0 to run in 32–bit mode. Refer to the Web Server documentation for instructions.

Cannot run `asant deploy`**: "The input line is too long" (Windows 2000) (ID 6485174)**

Description

When installing Application Server 8.2 in the default location on Windows 2000, you may encounter the following error when running `asant deploy`:

```
$ C:/Sun/JavaES5/appserver/bin/asant deploy
The input line is too long.
The syntax of the command is incorrect.
```

The problem is that command lines in Windows 2000 can be no longer than 1000 characters, and depending on your system configuration, the default ANT_OPTS environment may cause the asant deploy command line to be long. This is only an issue on Windows 2000.

Solution

On Windows 2000, install Application Server in a very short directory path; for example, C:\JES5_AS).

JES5 b12, AS installation, in common.properties **wrong server instance** AppServer1 **(ID 6485254)**

Description

Using JES 5 b12 on Windows, if Application Server is selected at the top level of the select components installation panel, then the Node Agent subcomponent is also selected by default. The installation process subsequently creates a node agent and a server instance named AppServer1 that belongs to this node agent. This is the correct behavior.

However, if the Node Agent subcomponent is deselected, the installation process still creates an AppServer1 instance in the common.properties file for the domain; for example:

```
domain.name=domain1
appserver.instance=AppServer1
```

Subsequent attempts to deploy applications using asant will fail.

Solution

Edit the common.propeties file, replacing appserver.instance=AppServer1 with appserver.instance=server.

Documentation on getting a physical Connection from a wrapped Connection is no longer correct (ID 6486123)

As a result of other defects (possibly 6295215) the code provided in the "Obtaining a Physical Connection from a Wrapped Connection" in *Sun Java System Application Server Enterprise Edition 8.2 Developer's Guide* section of Chapter 11, "Using the JDBC API for Database Access," in *Sun Java System Application Server Enterprise Edition 8.2 Developer's Guide* is not correct. Specifically, the line:

```
Connection drivercon = ds.getConnection(con);
```

should now read:

```
Connection drivercon = ((com.sun.gjc.spi.DataSource)ds).getConnection(con);
```

Application Server does not support NFS. (6396045)

In this version of the software, Application Server does not support Network File System (NFS).

Solution

None.

J2EE Tutorial

To run the J2EE 1.4 Tutorial on the Sun Java System Application Server Enterprise Edition 8.2 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.

- 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.

Lifecycle Management

This section describes known lifecycle management issues and associated solutions.

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

Description

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

- minimum-delivery-interval is the minimal interval duration between deliveries of the same periodic timer.

- redelivery-interval-in-mills is the time the timer service will wait after a failed ejbTimeout before attempting redelivery.

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

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

Solution

Use the default values for these properties, as follows:

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

Values other than these defaults will generate an error.

Logging

This section describes known logging issues and solutions.

Setting debug statement for `access.failure` causes hanging in Application Server startup. (ID 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.

Logging location/Instance location have changed for JES3 Application Server. (ID 6189409)

Default logging and server instance locations have changed in Sun Java System 8.2 compared to 7.x.

For more information, refer to the *Sun Java System Application Server Enterprise Edition 8.2 Administration Guide* or the *Sun Java System Application Server Enterprise Edition 8.2 Upgrade and Migration Guide*.

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. (IDs 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

Asynchronous message listener behavior changed in `appclient` **from 8.0 to 8.1 Update 2. (ID 6198465)**

Description

Due to a recent change, when an asynchronous message listener is the only live thread in the `app-client` container, the remaining `appclient` virtual machine exists as a daemon. This behavior is a regression for past applications that perform asynchronous receives in ACC. This problem affects application clients that set a JMS message listener and exit the main thread.

Solution

Do not exit the main thread. Wait for the message listener to notify the main thread before terminating the main thread.

Monitoring

This section describes known monitoring issues and associated solutions.

Cannot change connector service and connector connection pool monitoring level. (ID 6089026)

Description

Using the Monitoring Level setting page, if you change either the Connector Service or Connector Connection Pool monitoring level to `LOW` or `HIGH` and then save, both are not changed in the `domain.xml` file for the domain. However, if you change the JMS Service monitoring level to `LOW` or `HIGH` and save, the values for Connector Service and Connector Connection Pool are also changed at the same time. This problem does not occur when running the equivalent commands from the command line.

Solution

Only use the JMS Service component on the Monitoring Level page to change monitoring levels.

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

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)

Monitoring MBean for an undeployed EJB module is not removed, even though all statistics under that monitoring name are moved. (ID 6191092)

Description

For example:

```
EJBModuleMonitorMap().size() = 1  eventhough ejb module is
undeployed EJBModuleMonitor().getName() = sqe_ejb_s1_01
```

This true for both EJB modules and applications. Both programmatically (through MBean API) and through asadmin list/get, an empty monitoring MBean still exists.

Diagnostics

```
asadmin list -m "server.applications" shows the following output:
server.applications.MEjbApp
server.applications.__ejb_container_timer_app
server.applications.adminapp
server.applications.admingui
server.applications.com_sun_web_ui
server.applications._export_install_nov-11_domains_domain1_applications
_j2ee-modules_sqe_ejb_s1_01
```

You can look at statistics:

```
bin/asadmin list -m "server.applications._export_install_nov-11_domains
_domain1_applications_j2ee-modules_sqe_ejb_s1_01"
server.applications._export_install_nov-11_domains_domain1_applications_
```

```
j2ee-modules_sqe_ejb_s1_01.SQEMessage
server.applications._export_install_nov-11_domains_domain1_applications_
j2ee-modules_sqe_ejb_s1_01.TheGreeter
```

Once you undeploy:

```
_export_install_nov-11_domains_domain1_applications_j2ee-modules_sqe_
ejb_s1_01
```

If you do a list command, you still see the application:

```
asadmin list -m "server.applications"
server.applications.MEjbApp
server.applications.__ejb_container_timer_app
server.applications._export_install_nov-11_domains_domain1_applications_
j2ee-modules_sqe_ejb_s1_01
server.applications.adminapp
server.applications.admingui
server.applications.com_sun_web_ui
```

but it does not contain any monitoring statistics:

```
asadmin list -m "server.applications._export_install_nov-11_domains_
domain1_applications_j2ee-modules_sqe_ejb_s1_01"
Nothing to list at server.applications.-export-install-nov-11-domains-
domain1-applications-j2ee-modules-sqe-ejb-s1-01.
```

To get the valid names beginning with a string, use the wildcard ('*') character. For example, to list the names of all the monitorable entities that begin with server, use list "server.*".

Solution

This is harmless. Module can be safely redeployed with out any problems. The root monitoring Mbean is not removed, but it is empty.

Persistence

This section describes known and associated solutions related to Java Data Objects and Container-Managed Persistence

JDO76018: Unable to flush persistent instances due to circular dependencies. (ID 6500961)

Description

This exception is thrown if a chain of foreign key dependencies between instances modified (or created) in a transaction results in a circular dependency in the database.

Solution

Split the original set of operations into multiple transactions.

PointBase

This section describes known and associated solutions related to PointBase.

Setting the isolation levels on a connection pool for an application causes exceptions in PointBase. (ID 6184797)

Description

For a JDBC connection pool pointing to a PointBase database installation, setting the transaction-isolation-level pool attribute to any value other than the default (`Connection.TRANSACTION_READ_COMMITTED`) causes an exception. However, setting this same parameter to non-default values for pools pointing to other databases does not throw an exception.

Solution

For a JDBC connection pool pointing to a PointBase database installation, do not attempt to set the `transaction-isolation-level`.

PointBase throws an exception if a network server and embedded drivers are used together. (ID 6204925)

Description

The bundled PointBase sometimes throws an exception if the network server driver and the embedded driver are simultaneously used.

Solution

Use either the embedded driver or the network server driver, but not both.

Upgrade problem where the default PointBase database is overwritten. (IDs 6264969, 6275448)

Description

When upgrading to Application Server Enterprise Edition 8.2, the Update release patch overwrites the Pointbase default database.

Solution

Recreate or re-enter any scheme or data that existed prior to the upgrade. If you deployed applications with CMP beans with the generate table option, you must undeploy or redeploy the application to have the tables regenerated.

Samples

This section describes known and associated solutions related to the sample code included with the Application Server 8.2 product.

On Windows, setup-one-machine-cluster hangs but works on Solaris; mqfailover **requires** Ctrl+C **to cancel and then must be re-run. (ID 6195092)**

Description

From *install_dir*\samples\ee-samples\failover\apps\mqfailover\docs\index.html, if you run the following commands:

- Console 1

  cd *install_dir*\samples\ee-samples asant start-mq-master-broker1

- Console 2

  cd *install_dir*\samples\ee-samples asant start-mq-cluster-broker1

- Console 3

  cd *install_dir*\samples\ee-samples asant start-mq-cluster-broker2

- Console 4

  cd *install_dir*\samples\ee-samples asadmin start-domain domain1

If you have already executed asant setup-one-machine-cluster-without-ha or asant setup-one-machine-cluster-with-ha for any other Enterprise Edition sample, then execute asant configure-mq otherwise execute asant setup-one-machine-cluster-and-configure-mq. In this case, the command appears to succeed:

```
start_nodeagent: [echo] Start the node agent cluster1-nodeagent
[exec] Command start-node-agent executed successfully.
```

But then the system hangs indefinitely.

Solution

None at this time. This problem similarly affects all Enterprise Edition samples that use this ant target on Windows. A workaround is to Ctrl+C out of the hung process and then rerun it.

Documentation does not explicitly state that you need to create JMS resources before running the MQ Failover Sample Application following the asadmin **deploy instructions. (ID 6198003)**

Description

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. (ID 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

Sample documents missing after upgrade from 8.0 Platform Edition to 8.2 Enterprise Edition

Description

After updating from Application Server Platform Edition 8.0 to Application Server Enterprise Edition 8.2, you may receive an HTTP 404 "File not found" error when attempting to access the samples page.

Solution

Copy the sample documents from the 8.0 domains to the 8.2 domains.

Samples fail at runtime when run in Sparse local zone. (ID 6460970)

Description

If Application Server Enterprise Edition 8.2 is installed in a Solaris Global zone, and a Application Server domain is subsequently installed in a Sparse local zone, you may encounter problems running the sample applications if the file permissions for the domain in the Sparse zone are not sufficiently open during the deployment process.

Solution

During the deployment process, make sure the Application Server can retrieve the client JAR file, xmsClient.jar, and copy it into the sample location, (/usr/SUNWappserver/appserver/samples/webservices/security/ejb/apps/xms/xmsClient.jar. This is normally done automatically by the sample harness, but it will fail if the permissions on xmsClient.jar are not open.

Security

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

Cannot run WebServiceSecurity applications on Enterprise Edition with J2SE 5.0. (ID 6183318)

Description

WebServiceSecurity applications cannot run with J2SE 5.0 because:

- J2SE 5.0 PKCS11 does not support UNWRAP mode
- J2SE 5.0 PKCS11 does not support RSA/ECB/OAEPWithSHA1AndMGF1Padding with PKCS11

The J2SE team has filed "CR 6190389: Add support for the RSA-PKCS1 and RSA-OAEP wrap/unwrap mechanisms" for this bug.

Solution

Use J2SE 1.4.2 with any other JCE provider (not the one included by default). Note that hardware accelerator support will not be present in this configuration.

SSL termination is not working. (ID 6269102)

Description

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.

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 while upgrading from Application Server Enterprise Edition 8 to Application Server Enterprise Edition 8.2. (ID 6165528)

Description

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

Solution

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

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

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 RedHat distributions.

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

Solution

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

▼ To use command line installation mode

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

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

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

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

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

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

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

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

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>
```

Port conflict after upgrading Application Server from JES2 to JES5

Description

When updating from Application Server 7.x to 8.2, you may encounter a port conflict between the old and new installations, most likely with the default ports of 8080 and 8181.

Solution

Change the ports used in Application Server 8.2 to resolve the port conflict.

Derby database used by samples script is created under wrong location. (ID 6377804)

Description

There are two aspects to this bug:

1. When sample application setup scripts that use the Derby database are run, the Derby database gets created under the current directory or under `<install_root>/bin`.

2. The sample `build` Ant script creates a `password.txt` file that stores the admin password file under current directory, which will not be writable in non-root and sparse zones scenarios.

Solution

1. *Derby database location* – Use the `--dbhome` option with the `start-database` command to create the database at the value specified for `--dbhome`. For example, the following is the `asadmin` command syntax for `start-database`.

```
start-database [--dbhost 0.0.0.0] [--dbport 1527] [--dbhome db_directory] [--echo=false]
[--verbose=false]
```

2. *Location of the* `password.txt` *file* – By design, the samples directory is expected to be writable since all the build commands include the creation of a `password.txt` file in that directory. Be sure to install a working copy of the samples in a writable location.

LoginException during upgrade from 8.0UR1PE to 8.2EE; upgrade process aborts (ID 6445419)

Description

This problem occurs when you run the upgrade installation using admin credentials other than the defaults.

Solution

When performing a side-by-side upgrade using the file based installer for 8.xPE to 8.2EE, use the following admin credentials for the new Application Server:

- **admin user:** admin
- **admin password:** adminadmin
- **master-password:** changeit

After performing the upgrade, you can change these passwords as needed.

Upgrade Tool fails to detect existing but invalid directory input for Source Directory field (ID 6460122)

Description

The Upgrade Tool fails to detect an existing but invalid directory input for Source Directory field, and gives the impression that directory configuration is correct.

The expectation is that an "Invalid directory" message should pop up when an incorrect path is entered for the Source Directory. An invalid directory message correctly pops up if `/opt/SUNWappserverEE81UR2/` is entered for the Source Directory. However, when `/opt/SUNWappserverEE81UR2/domains` is entered, the tool continues with the upgrade process without warning, even though the path is invalid. This issue is similar to ID 6440710, except that the behavior differs depending on the input value.

Solution

When upgrading from Application Server 7 or 8.x to Application Server 8.2, the source directory must first be seeded with the value recommended in the documentation: domain root for in-place and domain directory for side-by-side upgrades.

Should invalidate Admin user/password name for semi-column (;) character (ID 6473341)

Description

The Application Server Enterprise Edition 8.2 installation does not allow special characters in the admin user name. Domain creation will fail if any special characters are used. Note, however, that the admin password can have special characters.

Solution

When upgrading from Application Server 7 to Application Server 8.2, verify that the admin user name does not contain any special characters.

Steps are missing for successful side-by-side upgrade. (6898037)

After upgrading from the older version of Application Server, verify the following under the upgraded installation (*target-installation*) location:

1. Domain directory is in the source directory.

2. Node agent is under `node-agent` directory as created in the source installation.

   cd *target-installation*/nodeagents/*upgraded-node*/agent/config

3. Check the value of the `agent.das.port` property in the `das.properties` file before starting the node agent for the first time. Perform this check on the `das.properties` file on the upgraded installation. The value of the `agent.das.port` property must reflect the same value as the `jmx-connector` port defined in the `domain.xml` file on the source installation from which the domain is upgraded.

4. Check the value of the `agent.bind.status` property in the `nodeagent.properties` file before starting the node agent for the first time. Perform this check on the `nodeagent.properties` file on *target-installation*. The value of the `agent.bind.status` property must BOUND. Edit the `agent.bind.status` property in the `nodeagent.properties` file on the source installation.

If you are upgrading side-by-side on Windows:

After upgrading the domain, if the *upgraded-node* has not been created under *target-installation*/nodeagents directory as in the source installation:

1. Go to the *target-installation*/bin directory.

2. Manually create the node agent that has the same name as in the source installation with the admin port specified in the create-node-agent command.

3. Perform the additional steps above.

4. Start the node agent.

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. (ID 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. (ID 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. (ID 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-passthrough **Web Server 6.1 Add-On. (ID 6188932)**

Description

The Sun Java System Application Server Enterprise Edition 8.2 adds support for the
functionality provided by the auth-passthrough plugin function available with Sun Java
System Application Server Enterprise Edition 7.1. However, in Application Server Enterprise
Edition 8.2, the auth-passthrough plugin feature is configured differently.

The auth-passthrough 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-passthrough 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-passthrough` 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-passthrough` plugin.

In Application Server Enterprise Edition 8.2, 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 Enterprise Edition 8.2. 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 Enterprise Edition 8.2 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.

HTTP listener created with `--enabled=false` doesn't disable the listener. (ID 6190900)

Description

When an `httplistener` is created with the `--enabled=false` flag, the listener does not become disabled. The flag `--enabled` does not have any effect when used at the same time the listener is created.

Solution

Create the listener in an enable state, then disable it manually later.

Redeployment on Windows fails because `verify_file_user_exists_common` **does not get executed. (ID 6490227)**

Description

On Windows, when redeploying an application that creates a user before deployment, the `create-file-user` command may fail because `verify_file_user_exists_common` does not get executed (though it is called), and fails to notify that the user already exists. Execution of the `deploy` target halts at this point, and the deploy and undeploy fail.

Solution

First delete the file user(s) using the `keydel` target, and then execute `deploy` target again:

```
asant keydel
asant deploy
```