

Sun Java™ System Application Server Enterprise Edition Release Notes

Version 8.1 2005Q1

Part Number 819-0214

The Sun Java™ System Application Server Enterprise Edition 8.1 2005Q1 product greatly simplifies the task of creating and administering web services applications. It provides superior performance, clustering, and high availability features for scalable services that continue to operate despite software and hardware faults. The Application Server provides a development path for web services that simplifies the development process while providing uniquely flexible growth opportunities.

These Release Notes contain important information available at the time of release of Sun Java System Application Server 8.1 2005Q1. New features and enhancements, known issues and limitations, and other information are addressed here. Read this document before you begin using Application Server 8.1.

The most up-to-date version of these release notes can be found at the Sun Java System documentation web site: <http://docs.sun.com/db/prod/slappsrv#hic/>. 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.

This document contains the following sections:

- [About Application Server Enterprise Edition 8.1 2005Q1](#)
- [Known Issues and Limitations](#)
- [How to Report Problems and Provide Feedback](#)
- [Sun Welcomes Your Comments](#)
- [Additional Sun Resources](#)

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.

About Application Server Enterprise Edition 8.1 2005Q1

The Sun Java System Application Server Enterprise Edition 8.1 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 section includes:

- [What's New in the 8.1 Release](#)
- [Hardware and Software Requirements](#)
- [Standalone Version](#)
- [Related Documentation](#)

What's New in the 8.1 Release

The Sun Java System Application Server Enterprise Edition 8.1 2005Q1 implements many new and enhanced features, described in the following sections:

- [Enhancements in This Release](#)
- [J2EE Support](#)
- [High Performance](#)
- [Scalability](#)
- [High Availability](#)
- [JavaServer Faces 1.1 Support](#)

Enhancements in This Release

The Application Server Enterprise Edition 8.1 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.
- **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.
- **Java Web Services Developer Pack 1.6 (JWSDP) Plugin Support** – All JWSDP plugins are now supported. The JWSDP 1.6 can be downloaded for free from <https://jsecom16b.sun.com/ECom/EComActionServlet;jsessionid=41C4780A0C9535BF236929F04FFD4590>.
- **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.
- **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). This 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 8.1 Installation Guide* for the details on HADB requirements and limitations.

J2EE Support

The Sun Java System Application Server 8.1 2005Q1 supports the J2EE 1.4 platform. The following table describes the enhanced APIs available on the J2EE 1.4 platform.

Table 1 Major API changes on the J2EE 1.4 Platform

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

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.

High Availability

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.

JavaServer Faces 1.1 Support

The Sun Java System Application Server Enterprise Edition 8.1 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 UI components with a JSP custom tag library.

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

Hardware and Software Requirements

This section lists the requirements that must be met before installing the Sun Java System Application Server Enterprise Edition 8.1 product.

- [Platform Requirements](#)
- [Important Patch Information](#)
- [JDBC Drivers and Databases](#)
- [Configuring Oracle](#)
- [Configuring PointBase](#)
- [Web Servers](#)
- [Browsers](#)
- [High Availability Requirements and Limitations](#)
- [HADB File System Support](#)
- [Upgrading the Sun Java System Application Server](#)
- [Switching to J2SE 1.4.2](#)
- [Other Requirements](#)

Platform Requirements

The following table lists the operating systems that are supported for Sun Java System Application Server Enterprise Edition 8.1 2005Q1 product. Additionally, the minimum and recommended memory requirements are identified for installing and running the Application Server.

Table 2 Sun Java System Application Server 8.1 2005Q1 Platform Requirements

Operating System	Minimum Memory	Recommended Memory	Minimum Disk Space	Recommended Disk Space	JVM
Sun Solaris 8, 9, 10 (SPARC) Solaris 9, 10 (x86)	512 MB	1 GB	250 MB free	500 MB free	J2SE 1.4.2_06 J2SE 5.0
Redhat Enterprise Linux 2.1 Update 2, 3.0 Update 1	512 MB	1 GB	220 MB free	300 MB free	J2SE 1.4.2_06 J2SE 5.0
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 1.4.2_06J2SE 5.0

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

Solaris Patch Requirements

It is recommended that Solaris 9, 10 (x86, SPARC) users have the “Sun recommended patch cluster” installed. This patch cluster is available under “Recommended and Security Patches” here:

<http://sunsolve.sun.com/>

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://mirrors.kernel.org/redhat/redhat/linux/9/en/os/i386/RedHat/RPMS/compat-libstdc++-7.3-2.96.118.i386.rpm>

Important Patch Information

For the current list of required patches for Sun Java System Application Server Enterprise Edition 8.1 go to <http://sunsolve.sun.com> and select either “Patches” or “Patch Portal.” Follow the Sun Java System Application Server Enterprise Edition 8.1 links. As operating system patch requirements change and patches to Java Enterprise System components become available, updates will be made available on SunSolve, initially in the form of recommended patch clusters.

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 3 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 Sybase ASE 12.5.2 Microsoft SQL Server 2000 4.0 Service Pack 1
IBM	Type 2	IBM DB2 8.1 Service Pack 3+
PointBase	Type 4	PointBase Network Server 4.8

Table 3 J2EE-Compatible JDBC Drivers (*continued*)

JDBC Vendor	JDBC Driver Type	Supported Database Server
DataDirect	Type 4	Oracle (R) 8.1.7, 9i, 9.2.0.3 Sybase ASE 12.5.2 Microsoft SQL Server IBM DB2 8.1 Service Pack 3+
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

For more information about i-net Software, see:

<http://www.inetsoftware.de/>

For more information about DataDirect Technologies, see:

<http://www.datadirect.com/>

Configuring Oracle

Oracle JDBC drivers must be configured properly to be compliant with J2EE 1.4. Use the following configuration for Type 2 and Type 4 drivers:

1. Use the JDBC driver from 9.2.0.3 or later.
2. The Oracle database needs to have `compatible=9.0.0.0.0` or higher in its parameter (`init.ora`) file.
3. Use the `ojdbc14.jar` file.
4. Configure the Application Server to define the following JVM property:

```
-Doracle.jdbc.J2EE13Compliant=true
```

In addition, for Type-2 drivers, both the `ORACLE_HOME` and `LD_LIBRARY_PATH` variables (which must include `$ORACLE_HOME/lib`) need to be defined in the environment in which the Application Server is started. For example, add them to the `asenv.conf` file and ensure they are exported.

Configuring PointBase

Many sample applications use the PointBase database server included with the Application Server. When using Application Server Enterprise Edition, you must configure the PointBase database server before using it. Before using PointBase with the Application Server, however, note the supported configuration combination.

Table 4 Supported J2SE/PointBase Combinations

Application Server	PointBase
Supported	
J2SE 1.4	J2SE 1.4
J2SE 5.0	J2SE 1.4
Unsupported	
J2SE 5.0	J2SE 5.0

There are two ways to configure PointBase:

- Set the `JAVA_HOME` environment variable to the location of the J2SE. The PointBase implementation bundled with Application Server 8.1 is only supported with J2SE 1.4.2.
- Edit the Application Server's PointBase configuration file.

To use the first method:

1. Make sure you have the J2SE installed that you want to use.

Download J2SE 1.4.2 if you do not already have it.

2. Using the command appropriate for your operating system and shell, set the `JAVA_HOME` environment variable to the directory in which J2SE is installed; for example:

```
% setenv JAVA_HOME "/opt/SUNWappserver/jdk"
```

To use the second method, the procedure depends on the operating system.

Solaris and Linux

Edit the `install_dir/pointbase/tools/serveroption/pbenv.conf` configuration file, changing the line:

```
PB_JAVA=%%PB_JAVA%%
```

to

```
PB_JAVA=J2SE_location
```

where *J2SE_location* is the directory where the J2SE is installed. If you installed J2SE with Application Server, it is installed by default to *install_dir*/jdk. After making this change, you can start PointBase using the `startserver` script.

Windows

Edit the *install_dir*\pointbase\tools\serveroption\pbenv.bat configuration file, changing the line:

```
set PB_JAVA=%%%PB_JAVA%%%
to
set PB_JAVA=J2SE_location
```

where *J2SE_location* is the directory in which the J2SE is installed. If you installed J2SE with Application Server, it is installed by default to *install_dir*\j2se1.4. After making this change, you can start PointBase by running `startserver.bat`.

Web Servers

This section lists the web servers that are supported for the Sun Java System Application Server Enterprise Edition 8.1 2005Q1.

Table 5 Supported Web Servers

Web Server	Version	Operating System
Sun Java System Web Server	6.1+	Solaris SPARC 8, 9, 10 Solaris x86 9, 10 Red Hat Enterprise Linux 2.1 Update 2, 3.0 Update 1
Apache Web Server	1.3+, 1.4, 2.0	Solaris SPARC 9, 10 Solaris x86 10 Red Hat Enterprise Linux 2.1 Update 2, 3.0 Update 1 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 8.1 2005Q1.

Table 6 Supported Web Browsers

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

High Availability Requirements and Limitations

The following high availability requirements must be met before configuring the Sun Java System Application Server High Availability component:

- HADB requires 512 MB minimum memory and 1GB recommended memory to work properly with the Application Server.
- HADB supports IPv4 only.
- The network must be configured for UDP multicast.
- Do not use dynamic IP addresses (DHCP) for hosts used in `create domain`, `extend domain`, `hadbm create`, or `hadbm addnodes` commands.
- If running HADB on Red Hat Linux 3.0, you must install Update 4 to avoid problems with excessive swapping by the operating system. Please note that AS has not been tested with Red Hat Update 4.

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 8.1.

Refer to the *Sun Java System Application Server Enterprise Edition 8.1 Installation Guide* for detailed information about installing and configuring HADB with Application Server 8.1 software.

Upgrading the Sun Java System Application Server

Refer to the Installation Guide for complete instructions for upgrading from a previous version of the Application Server to the Sun Java System Application Server Enterprise Edition 8.1 2005Q1.

Switching to J2SE 1.4.2

Sun Java System Application Server 8.1 2005Q1 supports J2SE 5.0 as the underlying JVM, however the bundled PointBase database does not. If you want to use PointBase with the Application Server, download J2SE 1.4.2 and use it instead of the bundled J2SE 5.0 JVM. To do this, perform the following steps:

1. Download the J2SE 1.4.2 SDK (not the JRE) from <http://java.sun.com/j2se/1.4.2/> and install it on your system, if you have not already done so.
2. Completely stop the Application Server.

You can use the following command line:

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

or 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 J2SE 1.4.2 home directory:
 4. Edit the `as-install/samples/common.properties` file, changing the line beginning `"com.sun.aas.javaRoot..."` to reference the J2SE 1.4.2 home directory.
 5. Restart the Application Server.

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

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 8.1 installation process. This allows the installation program to detect ports that are in use and avoid assigning them for other uses.
- **Replacing previously-installed servers (UNIX)** — 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 Upgrade and Migration Guide available at:

<http://docs.sun.com/app/docs/doc/819-0083>

Standalone Version

The standalone version of Sun Java System Application Server Enterprise Edition 8.1 differs in several ways from the Java ES Enterprise Edition version; specifically:

- The 8.1 standalone product can be installed by any user, whereas Java ES can only be installed as root user.
- The HADB component is visible as a subcomponent in the standalone version, whereas in the Java ES installation it is a shared component.
- The standalone version installs all shared components required for the Application Server under one installation directory, whereas in JES these components are installed in different directories.
- Product files, domains, and configuration data for the Application Server are stored by default in a single directory with the standalone installer, whereas with Java ES they are stored in multiple directories.
- The standalone version allows installation on a system with an already existing Application Server installation of the same or different version without the need to uninstall the existing installation. This is achieved by maintaining unique installation directories across versions OR across instances of the same version.
- The standalone version supports “upgrade in place” of an existing Sun Java System Application Server Platform Edition 8.0 installation or a Sun Java System Application Server Platform Edition 8.1 installation to the Sun Java System Application Server Enterprise Edition 8.1.

Related Documentation

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

http://docs.sun.com/app/docs/coll/ApplicationServer8_pe_04q4

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

Table 7 Books in This Documentation Set

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

Known Issues and Limitations

This section describes known problems and associated workarounds for the Sun Java System Application Server Enterprise Edition 8.1 2005Q1 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](#)
- [Application Client](#)
- [Bundled Sun JDBC Drivers](#)
- [Connectors](#)
- [Container Managed Persistence](#)
- [Deploytool](#)
- [Documentation](#)
- [High Availability](#)
- [Installation](#)
- [J2EE Tutorial](#)
- [Lifecycle Management](#)
- [Logging](#)
- [Message Queue](#)
- [Monitoring](#)
- [PointBase](#)
- [Samples](#)
- [Security](#)
- [Upgrade Utility](#)
- [Web Container](#)

Administration

This section describes known administration issues and associated solutions.

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

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

Solution

Do one of the following:

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

Cannot restore backed-up domain with another name. (ID 6196993)

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

Solution

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

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

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

Example values include:

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

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

Solution

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

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

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

For more information, refer to the *Sun Java System Application Server 8.1 Administration Guide*.

Overly restrictive execute permissions on Application Server start and stop scripts. (UNIX only) (ID 6206176)

If you run the `asadmin restore-domain` command while logged in as user "A", the scripts will end up with permissions as 744 (`rxwxr--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/*
```

Application Client

This section describes known application client issues and associated solutions.

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

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

Solution

None at this time.

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 *Application Server 8.1 2005Q1 Administration Guide* for details about configuring connection pools.

Solution

None.

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:

```
[sum][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 *Application Server 8.1 2005Q1 Administration Guide* for details about configuring connection pools.

Description 2

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

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

Solution 2

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

Description 3

Isolation level `TRANSACTION_SERIALIZABLE`.

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

Solution 3

To set desired isolation level for a connection, the corresponding connection pool has to be created at that isolation level. See the *Application Server 8.1 2005Q1 Administration Guide* for instructions.

On Application Server Enterprise Edition 8.1, the bundled DB2 Sun JDBC driver does not work in the default configuration. This happens because the DB2 JDBC driver classes require an explicit `charsetProvider` `RuntimePermission`. (ID 6183492)

Solution

Modify the `server.policy` file to provide the following permissions to a deployed application that uses the JDBC driver:

```
grant codeBase "file:${DEPLOYED_APPLICATION_DIR}" { permission java.lang.RuntimePermission  
"charsetProvider";  
};
```

Note that this is only required for the bundled DB2 Sun JDBC driver.

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

- 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 'sybssystemprocs.dbo.spt_server_info'.
```

Solutions

None.

Connectors

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

connection-validation is not dynamically reconfigurable in jdbc-connection-pools. (ID 4930792)

After a JDBC connection pool is created, its `is-connection-validation-required` attribute is not dynamically reconfigurable. This implies that for an already created pool connection, validation cannot be switched on (or off) on the fly. This is also true for the `validation-method` attribute of the pool.

Solution

In decreasing order of intrusiveness to running applications there are three possible workarounds:

1. Create `jdbc-connection-pools` with validation switched on.

2. Delete the `jdbc-connection-pool` and recreate it with `validation` switched on.
This will only affect at the most a few deployed applications that depend upon that particular pool.
3. Change the `validation` property and restart the Application Server.
This will affect all deployed applications (since there is a restart).

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

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

Container Managed Persistence

This section describes known container managed persistence issues and associated solutions.

An EJBQL query may not contain all matching results if the `where` clause contains an `OR` operator and a single-valued `cmr` navigation. (ID 6184864)

If the `where` clause in an EJBQL query contains an `OR` operator and a single-valued `cmr` navigation, the query result will not contain the result for rows in which the navigation path is `null` even though the navigation path is in a different `OR` clause.

For example, consider a schema comprising `Employee`, `Department`, and `Insurance`. `Employee` has a 1:Many relationship with `Department` and a 1:1 relationship with `Insurance`:

```
select Distinct Object(e) from Employee e
       where e.name = 'John' OR e.department.name = 'Engineering'
```

The above query will not return employees whose name is `John` and does not belong to any department.

```
select Distinct Object(e) from Employee e
       where e.department.name = 'Engineering' OR e.insurance.name = 'xyz'
```

The above query will not return any employee whose insurance name is `xyz` and does not belong to any department. It will also not return any employee whose department name is `Engineering` and does not have any insurance.

Solution

Execute the query for each `OR` condition separately and merge the results.

Deploytool

This section describes known Deploytool issues and associated solutions.

Deploytool often will not create `message-destination` elements in the following Sun deployment descriptors: (ID 6197393)

- `sun-application-client.xml`
- `sun-ejb-jar.xml`
- `sun-web.xml`

A JMS destination resource specified as the JNDI Name in the Message Destinations tab may not be saved to the Sun descriptor. After specifying the Destination Name (for example, `PhysicalQueue`, a physical destination created with `create-jmsdest`) and pressing Enter, the Destination Name appears under Display Name, and the client or bean name appears in the Producers list. After typing `"/Queue"` in the Sun-specific JNDI Name text field and pressing Enter, the application does not show as `"(changed)"` in the title bar, and an error is written to `~/ .deploytool/logfile`. When saving the application and going back to the tab, the JNDI Name field is blank again. When viewing the Sun descriptor using `Tools>Descriptor Viewer>Application Server Descriptor`, the `<message-destination>` element within the `<jndi-name>` element has not been created.

The problem is that during a deploytool session, the first time a value is entered for a Message Destination JNDI Name, the value appears correct in the Sun descriptor but an `IllegalArgumentException` is thrown by `org.netbeans.modules.schema2beans.BeanProp.setElement()`. Subsequent changes or additions of a Message Destination JNDI Name in the same application or other applications will not be saved to the Sun descriptor.

Solution

To edit an existing JNDI Name of a Message Destination:

1. Delete the existing JNDI Name by leaving the JNDI Name text field blank and pressing Enter.
2. Type the new JNDI Name and press Enter.

3. Review the Sun descriptor by clicking Tools>Descriptor Viewer>Application Server Descriptor.
4. Save the application by clicking File>Save.

If the JNDI Name is not saved to the Sun descriptor:

1. Restart deploytool.
2. On the Message Destinations tab, select a Message Destination or add a new Message Destination.
3. Enter the JNDI Name for the Message Destination in the Sun-specific JNDI Name text field, and then press Enter.
4. Review the Sun descriptor by clicking Tools>Descriptor Viewer>Application Server Descriptor.
5. Save the application by clicking File>Save.

Repeat the above steps each time a value needs to be entered in the Sun-specific JNDI Name on the Message Destinations tab, unless a value is being entered in the JNDI Name text field for the first time during a deploytool session.

Broken panels in the New Web Service Wizard (ID 6198981)

This problem manifests with two sets of symptoms:

- Clicking Next on the WSDL File panel does not go to the next panel, and an exception is thrown:

```
java.lang.NoClassDefFoundError: com/sun/org/apache/xpath/internal/XPathAPI
```
- On Create Packaged Endpoint panel, select Create Web Service Endpoint and Package. Clicking Finish on the Next Steps panel does not create the Endpoint module in the output directory, and an exception is thrown:

```
java.lang.NoClassDefFoundError: com/sun/org/apache/xpath/internal/XPathAPI.
```

The problem is that `xalan.jar`, which contains the `XPathAPI.class`, is missing from the CLASSPATH. Note that this problem does not exist with JDK 5.0.

Solution

Add `xalan.jar` to the CLASSPATH includes for the `slas-deploytool` process of `install_dir/lib/processLauncher.xml`; for example:

```
includes="appserv-assemblytool.jar,activation.jar,appserv-admin.jar,appserv-cmp.jar,appserv-rt.jar,j2ee.jar,jaxrpc-impl.jar,appserv-ext.jar,deployhelp.jar,admin-cli.jar,dom.jar,xercesImpl.jar, xalan.jar"
```


“Home” incorrectly translated as “installation directory” in Deploytool for Simplified Chinese. (ID 6203658)

When you create an Enterprise Bean in deploytool, and then navigate to the Transaction or Security tab for the bean node, the “Local Home” and “Remote Home” labels are incorrectly translated as “Local Installation Directory” and “Remote Installation Directory.”

Documentation

This section describes known documentation issues and associated solutions.

Errors in index.html and QuickStart.html documents (ID 6193749)

There are two sets of errors in the `index.html` file for the Application Server 8.1 documentation set and the `docs-ee/QuickStart.html` file.

1. The default `index.html` page copied to the `docroot` directory for each domain shows the incorrect path; it should be:

```
as_instance/domains/domain_name/docroot/
```

not

```
install_dir/domains/domain_name/docroot/index.html
```

2. The QuickStart Guide provides incorrect installation instructions for load balancer and web servers. For updated instructions, see the online QuickStart Guide at:

<http://docs.sun.com/app/docs/doc/819-1003>

Note that this affects the Java ES Enterprise Edition version of Application Server 8.1 only, and does not affect the [Standalone Version](#).

The `- asadmin create-domain --help` command produces incorrect usage and an invalid option is documented (`--admin.jmxport`). (ID 6207862)

The help command for `asadmin create-domain` describes `--admin.jmxport`, which is not a valid option for this command.

Solution

The `--admin.jmxport` cannot be used with the `asadmin create-domain` command.

AppservPasswordLoginModule referenced as AbstractPasswordLoginModule in documentation (ID 6229682)

The “Realms” section in Chapter 2, “Securing Applications,” in the *Sun Java System Application Server Enterprise Edition 8.1 2005Q1 Developer’s Guide* incorrectly refers to extending `com.sun.appserv.AbstractLoginModule`, however this class is now named `com.sun.appserv.AppservLoginModule`.

Solution

Refer to `com.sun.appserv.AppservLoginModule` instead of `com.sun.appserv.AbstractLoginModule`.

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.

High Availability

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

The requirements for using Apache with Sun Java System Application Server in the “Compiling and Configuring Apache Web Server” Appendix of the *Administration Guide* are out of date.

Listed below are the software requirements for using Apache web server software with HADB.

- **Requirements for Apache 1.3**
 - openssl-0.9.7e (source)
 - mod_ssl-2.8.16-1.3.33 (source)
 - apache_1.3.33 (sources)
 - gcc-3.3-sol9-sparc-local packages (for Solaris 9 SPARC/ x86)
 - gcc-3.3-sol9-intel-local packages (for Solaris 9 x86)
 - flex-2.5.4a-sol9-sparc-local packages (for Solaris 9 SPARC)
 - flex-2.5.4a-sol9-intel-local packages (for Solaris 9 x86)

- **Requirements for Apache 2.0**
 - openssl-0.9.7e (source)
 - httpd-2.0.49 (source)
 - gcc-3.3-sol9-sparc-local packages (for Solaris 9 SPARC).
 - gcc-3.3-sol9-intel-local packages (for Solaris 9 x86)
 - flex-2.5.4a-sol9-sparc-local packages (for Solaris 9 SPARC)
 - flex-2.5.4a-sol9-intel-local packages (for Solaris 9 x86)

There is also an additional step required before compiling. On the Solaris 10 platform, before running `make` for OpenSSL, run the following command:

```
/usr/local/lib/gcc-lib/sparc-sun-solaris2.9/3.3/install-tools/mkheaders.
```

Finally, the information in “Configuring Load Balancing and Failover” in the *Administration Guide* detailing modifications you must make to the Apache web server after installation is incomplete.

On All Platforms

1. Create a directory called `sec_db_files` under `apache_install_dir`.
2. Copy `domain/config/*.db` to `apache_install_dir/sec_db_files`.

On the Solaris Platform:

- Add the path `/usr/lib/mps/secv1` to `LD_LIBRARY_PATH` in the `apache_install_dir/bin/apachectl` script. The path must be added before `/usr/lib/mps`.

On the Linux Platform

- Add the path `/opt/sun/private/lib` to `LD_LIBRARY_PATH` in the `apache_install_dir/bin/apachectl` script. The path must be added before `/usr/lib`.

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).

New tables created after new nodes are added are not fragmented on the added nodes. (ID 5042351)

If you create a database instance and then add nodes to it, any new tables created afterwards will not be fragmented on the nodes added after database creation. Only the tables created before the `addnodes` command will be able to use the added nodes when `hadbm addnodes refragment` it. This is because `create table` uses the `sysnode` node group that is created at when `hadbm create` is executed.

Solution

Run `hadbm refragment` after any new tables are added, or create the new tables on the node group `all_nodes`.

Heterogeneous paths for `packagepath` not supported. (ID 5091349)

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 europall
Package successfully registered.
hadbm registerpackage test --packagepath=/var/install2 --hosts europal2
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.

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

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.

HADB problem with RedHat AS 3.0 in co-located mode under load. (ID 6158393)

HADB runs on RedHat 3.0 co-located with AS. Transactions may get aborted and affect the performance. This is caused by the excessive swapping performed by the operating system.

Solution

The problem has been fixed in Red Hat EL 3.0 Update 4. HADB has been tested with RedHat 3.0 update 4, and it is verified that excessive swapping by the operating system has disappeared. Note that Application Server 8.1 has not been tested with update 4.

The `configure-ha-cluster` command may hang. (ID 6159633)

When the `asadmin configure-ha-cluster` command is used to create or configure a highly available cluster on more than one host, the command sometimes hangs. There are no exceptions thrown from the HADB Management Agent or the Application Server.

Solution

HADB does not support heterogeneous paths across nodes in a database cluster. Make sure that the HADB server installation directory and configuration directory are the same across all participating hosts. Be sure to clear the repository directories before running the command again.

Application Server Performance with HADB (ID 6172589)

On all platforms, performance of Application Server instances configured to use HADB will be worse than the previous release due to changes to the JDBC drivers used by HADB.

Solution

Contact Sun Services immediately for resolution.

Second addnodes fails during refragmentation. (ID 6175436)

The second (and subsequent) addnodes command may fail during refragmentation with the following error:

```
hadbm:Error 22042: Database could not be refragmented. Please retry with hadbm refragment
command to refragment the database. Caused by: HADB-E-11747: Nodegroup all_nodes exists
already
```

Solution

Refragment the tables manually using hadbm refragment.

Cannot create a data device larger than 2GB on Windows. (ID 6181845)

If you use hadbm create or hadbm set with --NumberOfDataDevices=1 (default) and --devicesize with a value larger than 2GB, the following error occurs:

```
DEVINIT-ERROR: out of space, wrote -2147479552 B of -2036330496 B
An attempt was made to move the file pointer before the beginning of the file.
```

Solution

If you need to create a data device larger than 2GB in Windows, divide the devicesize by 2GB and find the number of devices you need. Then create many data devices according to the calculation, using the --NumberOfDataDevices option. For example, if you need to create a 5GB data device:

$$5 / 2 = 2.5$$

You should round up and set --NumberOfDataDevices=3.

Information in hadbm help is outdated (ID 6190702)

Some information in the hadbm help system is outdated.

Solution

See the HADB chapter in the *Application Server 8.1 Administrator's Guide* for the latest information.

Addnodes command fails with table not found error (ID 6214601)

In this scenario, the hadbm refragment command fails with the following error:

```
hadbm:Error 22042: Database could not be refragmented. Please retry with hadbm refragment
command to refragment the database.. Caused by: HADB-E-11701: *Table singlesignon not
found*
```

Solution

The workaround is to refragment the App Srv tables manually by using the clusql command:

```
> clusql <server:port list> system+<dbpassword specified at database create>
SQL: set autocommit on;
SQL: set schema haschema;
SQL: alter table sessionattribute nodegroup all_nodes;
SQL: alter table singlesignon nodegroup all_nodes;
SQL: alter table statefulsessionbean nodegroup all_nodes;
SQL: alter table sessionheader nodegroup all_nodes;
SQL: alter table blobsessions nodegroup all_nodes;
SQL: quit;
```

The management agent terminates with the exception "IPV6_MULTICAST_IF failed" (ID 6232140)

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.

Installation

This section describes known installation issues and associated solutions.

Intermittent failure to render “Next” navigation button on installer and uninstaller Welcome screen. This issue only affects the **Standalone Version** of the product. (ID 4977191)

This problem has been reported intermittently on the Solaris x86 platform, but it is possible that it also affects Solaris SPARC and Linux platforms.

The problem is that the installer's or uninstaller's first screen correctly displays the full text and “Help” and “Cancel” buttons, but the “Next” button necessary to navigate to the next screen is not visible. Although button is not visible, its area is active and if you click on it, navigation to the next screen proceeds normally. The cause of the problem is intermittent J2SE GUI repaint issue.

Solution

One workaround is to click on the “Next” button area just to the left of the “Help” button. Another workaround is to force repainting of the screen by resizing it slightly or by minimizing and restoring the installer window. After repainting, the missing “Next” button will become visible.

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

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

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

Solution

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

file://install_dir/docs-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.

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

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

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

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

Solutions

To work around this issue:

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

The `imq` directory needs to be created during installation (Windows only). (ID 6199697)

Immediately after installing Application Server EE on Windows, IMQ broker fails on startup with a message saying the directory `drive:\astomains\omain1\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

Create the `var_home_dir_location` before creating the broker:

```
$imqbrokerd -varhome var_home_dir_location
```

For example:

```
$imqbrokerd -varhome D:\astomains\omain1\imq
```

J2EE Tutorial

To run the J2EE 1.4 Tutorial on the Sun Java System Application Server Enterprise Edition 8.1 2005Q1 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 Admin 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-millis` to 7000 causes the `set` command to fail with the following error: (ID 6193449)

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

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

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

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

Solution

Use the default values for these properties, as follows:

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

Values other than these defaults will generate an error.

Logging

This section describes known logging issues and solutions.

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

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

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

Solution

None at this time. Please avoid setting this flag.

Message Queue

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

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

Failures to reconnect in timing-dependent scenarios can be caused by several problems. In general, however, 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 (ID 6198465)

Due to a recent change, when an asynchronous message listener is the only live thread in the `app-client` container, the remaining `appclient` VM 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.

Message Broker logs contain “unable to deliver” messages. (ID 6204180)

When running in a clustered environment, if a broker is low in memory, the following error messages may be seen in the broker log:

```
Internal error, unable to deliver .....: java.lang.NullPointerException
```

With durable consumers, although the message has been acknowledged by a consumer, the message may be redelivered to consumers (with the redeliver flag set) at a later time.

This error occurs because the data that was still needed by the system was incorrectly freed when the memory on the system became constrained. This error only occurs on messages that are delivered to consumers attached to this broker but which were propagated to a different broker in the cluster.

Solution

Increase the maximum Java heap size for the Message Broker Process (-Xmx) to prevent the system from running low on memory.

Broker running in a cluster runs out of memory after restart after a failure. (ID 6205463)

A Message Broker running as part of a cluster runs out of memory after it has been restarted. There are two different issues which may cause this problem. To determine if one or both are relevant, examine the log files for Application Servers and Message Brokers in the cluster:

- If the Application Server log file has WARNING messages with the format:

```
MQRA:OMR:Endpoint Unavailable:Marking OMR as invalid-omrId=1 for:ActivationSpec  
configuration ...
```

then the memory issue is being triggered as described in [“JMS reconnection does not successfully complete in certain cases that are timing dependent. \(ID 6173308, 6189645, 6208728, 6198481, 6199510, 6199510\)”](#).

- If one of more of the broker log files on other brokers in the cluster show messages of the format:

```
ERROR Internal Error: received ack twice on ...
```

then the memory issue is being triggered as described in [“Memory buildup on message broker when cluster is restarted after a failure. \(ID 6208621\)”](#).

Solutions

See the solutions listed for 6208621 or 6208728 (depended on which issues are causing the problem).

Memory buildup on message broker when cluster is restarted after a failure. (ID 6208621)

When a Message Broker in a cluster is restarted after a failure, memory may build up on the broker because the state of non-durable MDBs on a Topic Destination are not correctly propagated.

After a Message Broker is started as part of an active cluster, exceptions are seen in both the broker and application server logs. Over time, the restarted broker will begin to run low on memory. In such cases, one or more of the logs for the application servers in the cluster will throw a WARNING message similar to the following after the broker restarts:

```
[#|2004-12-03T17:45:12.821-0800|WARNING|sun-appserver-ee8.1|javax.enterprise.system.stream
.err|_ThreadID=12;|com.sun.messaging.jms.JMSEException: [C4000]: Packet acknowledge failed.
user=admin, broker=<brokername>....
```

A corresponding message is also usually seen in the Message Broker logs, and has the format:

```
ERROR Internal Error: received ack twice on ...
```

The problem is that when a Message Broker attaches to an active cluster, information about all active consumers are forward to it by other brokers in the cluster. If an MDB has a non-durable subscriber on one of the remote brokers, it may send incorrect information when it forwards the consumer information. Once the invalid consumer information is received by the restarted broker, that broker will incorrectly route extra copies of a message to the other broker.

When this occurs, the remote consumer will log a "[C4000]: Packet acknowledge failed" message in the ApplicationServer log and a "double ack" error will be logged on the remote broker. Each time a "double ack" message is seen in the broker log, that message will not be correctly acknowledged by the producing broker. Over time, this causes the producing broker to run low or out of memory.

Solutions

If the problem is currently occurring, the MDB can be undeployed and redeployed across the system to clean up the internal information. To prevent the issue from affecting the operation of the broker during normal operation:

- If performance is not an issue, change the MDBs to use durable (instead of non-durable) subscribers.

- If performance is an issue, configure the system so that the “leaked” messages are cleaned up if an acknowledgement is not received within a period of time. This can be done by either:
 - Setting a `limitBehavior` of `REMOVE_OLDEST` and a message limit on the destination used by the MDBs:

```
imqcmd update dst -n <dest name> -t t -o limitBehavior=REMOVE_OLDEST -o
maxNumMsgs=<some limit>
```
 - Change “time to live” setting on messages that are propagated to the topic.

Monitoring

This section describes known monitoring issues and associated solutions.

Some of the HTTP Service monitoring statistics do not present useful information and should be ignored. (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 8.1, and should be ignored:

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

Solution

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

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

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 MBeanAPI) 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_e
jb_s1_01
```

You can look at statistics:

```
bin/asadmin list -m "server.applications._expo
rt_install_nov-11_domains_domain1_applications_j2ee-modules_sqe_ejb_s1_01"
server.applications._export_install_nov-11_domains_domain1_applications_j2ee-mod
ules_sqe_ejb_s1_01.SQEMessage
server.applications._export_install_nov-11_domains_domain1_applications_j2ee-mod
ules_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-mod
ules_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.

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)

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)

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.

Samples

This section describes known and associated solutions related to the sample code included with the Application Server 8.1 product.

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

For example, to reproduce the error, see

`install_dir\samples\ee-samples\failover\apps\mqfailover\docs\index.html`, and then 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 EE 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 EE 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)

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.

Solutions

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

Runtime error during certificate creation in web services/security samples on Linux. (ID 6198239)

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.

Solutions

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=$SAS_NSS:$LD_LIBRARY_PATH;export LD_LIBRARY_PATH
```

The documentation for the `- ee samples asadmin deploy` command omits the `availabilityenabled=true` option for deploying the application, which ensures the sample applications are HADB-enabled by default. (ID 6198796)

Solutions

Using the `asadmin` command:

1. Go to the root of the sample directory; for example:

```
cd install_dir/samples/ee-samples/failover/apps/dukesbookstore
```

2. Execute `asadmin deploy` to deploy the application to the local Application Server instance; for example:

```
install_dir/bin/asadmin deploy -user adminuser -password adminpassword --host DASHost --port DASport --availabilityenabled=true --target cluster-name dukesbookstore.ear
```

Do the same thing for the `asadmin deploy` commands for all other EE samples, with the exception of `install_dir/samples/ee-samples/failover/apps/mqfailover`. Note that MQ does not use HADB.

Cannot run failover test with the asant script for the dukesbookstore EE sample. (ID 6199076)

After setting up a two-machine cluster, the `dukesbookstore failover` sample encounters errors. The idea here is to have one database per cluster. Currently, when you deploy a sample from DAS on a cluster with instances running on two separate machines, the scripts use the PointBase host as `localhost`. When an EE sample is deployed, the JDBC resources get deployed with the PointBase host as `localhost` on both the instances. Therefore, `localhost:9092` on both instances expect PointBase to be running on both machines.

The problem is that two instances belonging to one cluster cannot use different databases. To get past this problem, if you replace `localhost` in `database.properties` with a host name, both instances of the cluster will be able to access the database: one through `localhost`, the other through the host name you specify.

Solutions

Edit the `install_dir/samples/database.properties` file on the host on which the PointBase server is running, setting the value for `pointbase.server` to the host name on which PointBase is running instead of `localhost`.

The current setup—that is, `pointbase.server=localhost` works for a one-machine cluster, but will not work for a two-machine cluster where PointBase may not be running on `localhost` for the second instance.

The MQ-failover sample application has the cluster name hard coded to “cluster1” in one of the Ant setup targets. (ID 6202363)

The MQ-failover sample application has `cluster1` hard-coded in one Ant setup target. Therefore, if you modify the `cluster-name` in `cluster.properties` in `ee-samples` from `cluster1` to a different name, the sample fails while trying to set the default host in `cluster1`:

```
set-default-jms-host-to-broker1:
[echo] Setting default JMS host to samplesbroker1
[exec] No object matches the specified name "cluster1-config.jms-service.default-jms-host"
[exec] CLI137 Command set failed.
```

The `install_dir/samples/ee-samples/build.xml` file hard codes reference to `cluster1`, when instead it should be using `${cluster.name}` from `install_dir/samples/ee-samples/cluster.properties`.

Solution

The hard-coded `cluster1` string must use `${cluster.name}` instead. Manually modify `install_dir/samples/ee-samples/build.xml` to change `set-default-jms-host-to-broker1` from `cluster1` to `${cluster.name}` or the customer cluster name specified in `cluster.properties`.

Security

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

Specifying target message by `java-method` does not work in `client-side message-security-binding` elements. (ID 6155080)

This problem occurs, for example, when a target message in a `client-side message-security-binding` element is specified by `java-method` within a `port-info` element within a `service-ref` element:

```

<!ELEMENT service-ref ( service-ref-name, port-info*, call-property*, wsdl-override?,
service-impl-class?, service-qname? )>
<!ELEMENT port-info ( service-endpoint-interface?, wsdl-port?, stub-property*,
call-property*, message-security-binding? )>
<!ELEMENT message-security-binding ( message-security* )>
<!ELEMENT message-security ( message+, request-protection?, response-protection? )>
<!ELEMENT message ( java-method? | operation-name? )>

```

The `message-security-binding` element is used here to define message protection policies for specific methods of a web service endpoint.

Solution

Use an `operation-name` element within the `message` element to identify by WSDL operation name the message to which the protection policies defined in the containing `message-security` element apply.

“CertificateNotYetValidException” if DAS and remote node-agent machines clocks not synchronized. (ID 6181989)

In cases where the DAS server and node-agents are installed on different machines, and the clocks on those machines are not synchronized, attempts to run the `asadmin --start remote-node-agent` command fail with a `CertificateNotYetValidException` error.

Solution

Synchronize the clocks on the DAS server and all remote node-agent machines.

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

WebServiceSecurity applications cannot be run with J2SE 5.0 because of the the following reasons:

- 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 communication with MQ does not work if `mq-scheme` and `mq-service` are set in `jms-serviceSSL`; communication with MQ does not work if `mq-scheme` and `mq-service` are set in `jms-service`. (ID 6202606)

The information used not set in the resource adapter to be used by connection factories created from it. So, connection factories created from it will not have SSL information.

Solution

If you need to use SSL communication between the Application Server and MQ, create the connection factory specifying the `addresslist` explicitly with SSL syntax. For example, `addresslist` in the connection factory could be:

```
mq://mqserver-1:7676/ssljms,mq://mqserver-2:7676/ssljms
```

SSL communication between the Application Server and MQ may be required when MQ and the Application Server are in different locations and the network connectivity between them can be exploited by an intruder.

URL to `https` listener specified without port number redirected to `http` listener at port 80 (ID 6207297)

When an SSL listener is enabled on the default port (443), specifying a URL in a browser to that secure port without also specifying the port number causes the browser to redirect to port 80 on the non-secure (`http`) listener.

For example:

1. Create an SSL listener on port 443 and restart the Application Server.
2. Point your browser to <https://servername:443>.

The page loads correctly

3. Point your browser to <https://servername> (no port number).

The browser loads <http://servername:80> instead of <https://servername>.

This problem does not occur when the SSL listener is on a port other than the default (443).

Solution

Choose either of two solutions:

- Advise site visitors and code all links to always specify the port number when accessing the SSL listener on `https` port 443.
- Do not use port 443 for any SSL listeners.

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 8.1. (ID 6165528)

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

Solution

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

During the upgrade from J2EE 1.4 SDK to Application Server EE 8.1, the bundled J2SE installation is not correctly upgraded. This issue only affects the [Standalone Version](#) of the product. (ID 6196741)

The problem occurs during the upgrade from the J2EE 1.4 SDK to the Application Server 8.1 standalone installation. In the course of this upgrade, the bundled J2SE 1.4.2 should be upgraded to J2SE 5.0. However, some JAR files in the resulting J2SE 5.0 installation are incorrectly upgraded, resulting in a corrupted J2SE 5.0 installation.

The installer reports a successful upgrade, and it is not expected that you will encounter any issues while running upgrade tool during the upgrade process. However, subsequent attempts to start the upgraded Application Server fails with the following exception:

```
Exception in thread "main"
[#|2004-11-17T18:12:24.033-0800|WARNING|sun-appserver-ee8.1|javax.enterprise.system.stream
.err|_ThreadID=10;|java.lang.NoClassDefFoundError: javax/net/ssl/TrustManager at
com.sun.enterprise.security.SecurityLifecycle.onInitialization(SecurityLifecycle.java:59)
at
com.sun.enterprise.server.ApplicationServer.onInitialization(ApplicationServer.java:215)
at com.sun.enterprise.server.PEMain.run(PEMain.java:277) at
com.sun.enterprise.server.PEMain.main(PEMain.java:219)
```

Solution

There are several workarounds for this problem:

- Install a standalone J2SE of the appropriate supported version. During the Application Server upgrade, choose the option to reuse the existing Java 2 SDK in the Application Server installer's "Java Configuration" screen instead of the default option to install Java 2 SDK 5.0. Provide the path to the standalone J2SE installation.

- Prior to running the upgrade, remove or rename the existing J2EE 1.4 SDK *install_dir*/jdk subdirectory. J2SE 5.0 will then be installed correctly during the Application Server 8.1 upgrade process. Please note that you should remove or rename this directory only after you have navigated through the installer's directory selection screen and have been prompted to allow "In place upgrade" of the existing installation.
- If the upgrade has already been performed and you have encountered the server startup problem, install a standalone J2SE of appropriate supported version, and then modify the AS_JAVA variable in the *install_dir*/config/asenv.conf file (Linux and Solaris) or the *install_dir*\config\asenv.bat file (Windows). The value of this variable should point to the location of the new standalone J2SE installation.

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

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

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

Solution

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

1. If you ran upgrade in place in GUI mode and encountered this problem, exit the installer by pressing Ctrl+C in the terminal window in which the installer was started.
2. Start upgrade tool from the terminal window, using following command:

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

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

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

file:///install_dir/docs-ee/about.html

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

Web Container

This section describes known web container issues and associated solutions.

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

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
```

Solutions

If you specify `--precompilejsps=false` (the default setting) when you deploy an app, 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)

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.

Using the AMX API, removing a J2EE application reference from a server removes the application, but the application is still accessible. (ID 6173248)

When using the AMX API, removing a reference to an application without first explicitly stopping the application results in that application still being accessible. This behavior is by design, and is a documentation omission.

Solution

To remove an application so it is no longer accessible:

1. Stop the application
2. Remove the reference to the application.
3. Undeploy the application.

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

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

```

at org.apache.tools.ant.taskdefs.Execute$Javal3CommandLauncher.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.

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** – A form for submitting feedback on the Application Server product, here:

<http://java.sun.com/docs/forms/J2EE14SubmittalForm.html>

- **J2EE-INTEREST list** – A mailing list for J2EE questions, here:

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

- **Bug database on Java Developer Connection** – To view bugs or to submit a bug, use the Java Developer Connection Bug Parade here:
<http://developer.java.sun.com/servlet/SessionServlet?url=/developer/bugParade/index.jshtml>
- **Java Technology Forums** – An interactive message board for sharing knowledge and questions about Java technologies and programming techniques. Use the J2EE SDK forum here for discussions related to the Sun Java System Application Server Enterprise Edition 8.1 product:
<http://forum.java.sun.com/>
- **Sun Software Support services** – online at:
<http://www.sun.com/service/sunone/software>
This site has links to the Knowledge Base, Online Support Center, and Product Tracker, as well as to maintenance programs and support contact numbers.
- The telephone dispatch number associated with your maintenance contract

So that we can best assist you in resolving problems, please have the following information available when you contact support:

- Description of the problem, including the situation where the problem occurs and its impact on your operation
- Machine type, operating system version, and product version, including any patches and other software that might be affecting the problem
- Detailed steps on the methods you have used to reproduce the problem
- Any error logs or core dumps

Sun Welcomes Your Comments

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

To share your comments, go to <http://docs.sun.com> and click Send Comments. In the online form, provide the document title and part number. The part number is a seven-digit or nine-digit number that can be found on the title page of the book or at the top of the document. For example, the title of this book is *Sun Java System Application Server Enterprise Edition 8.1 2005Q1 Release Notes*, and the part number is 819-0214.

Additional Sun Resources

Useful information can be found at the following locations:

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

Copyright © 2004-2005 Sun Microsystems, Inc. All rights reserved.

Sun Microsystems, Inc. has intellectual property rights relating to technology embodied in the product that is described in this document. In particular, and without limitation, these intellectual property rights may include one or more of the U.S. patents listed at <http://www.sun.com/patents> and one or more additional patents or pending patent applications in the U.S. and in other countries.

SUN PROPRIETARY/CONFIDENTIAL.

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

Use is subject to license terms.

This distribution may include materials developed by third parties.

Portions may be derived from Berkeley BSD systems, licensed from U. of CA.

Sun, Sun Microsystems, the Sun logo, Java and Solaris are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries.

Copyright © 2004-2005 Sun Microsystems, Inc. Tous droits réservés.

Sun Microsystems, Inc. détient les droits de propriété intellectuelle relatifs à la technologie incorporée dans le produit qui est décrit dans ce document. En particulier, et ce sans limitation, ces droits de propriété intellectuelle peuvent inclure un ou plusieurs des brevets américains listés à l'adresse <http://www.sun.com/patents> et un ou des brevets supplémentaires ou des applications de brevet en attente aux Etats - Unis et dans les autres pays.

Propriété de SUN/CONFIDENTIEL.

L'utilisation est soumise aux termes du contrat de licence.

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

Des parties de ce produit pourront être dérivées des systèmes Berkeley BSD licenciés par l'Université de Californie.

Sun, Sun Microsystems, le logo Sun, Java et Solaris sont des marques de fabrique ou des marques déposées de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays.

Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays.