

Sun Java™ System Application Server Platform Edition Release Notes

Version 8.1 2005Q1

Part Number 819-0072

The Sun Java™ System Application Server Platform Edition 8.1 2005Q1 product is a J2EE 1.4 platform-compatible server for the development and deployment of J2EE applications and Java Web Services. Production use of this server is free of charge. Sun Java System Application Server Platform Edition is free for development, deployment and redistribution. Customers interested in redistribution should contact Sun OEM sales for a redistribution license:

http://www.sun.com/software/products/appsrvr/appsrvr_oem.html

These Release Notes contain important information available at the time of release of Sun Java System Application Server07

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 Platform 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 Platform Edition 8.1 2005Q1

The Sun Java™ System Application Server Platform Edition 8.1 2005Q1 is a J2EE 1.4 platform-compatible server for the development and deployment of J2EE applications and Java technology-based web services.

This section includes:

- [What's New in the 8.1 Release](#)
- [J2EE Support](#)
- [Hardware and Software Requirements](#)
- [Switching to J2SE 5.0](#)
- [Related Documentation](#)

What's New in the 8.1 Release

The Sun Java System Application Server Platform Edition 8.1 2005Q1 implements many new features:

- **J2EE 1.4 Platform Support** – This release implements all of the specifications covered by the J2EE 1.4 platform.
- **JavaServer Faces 1.1 Support** – All levels of developers can quickly build web applications by assembling reusable UI components in a page, connecting these components to an application data source, and wiring client-generated events to server-side event handlers.

- **JavaServer Pages Standard Tag Library 1.1 Support** – The Application Server supports the JavaServer Pages Standard Tag Library (JSTL) 1.1, which encapsulates core functionality common to many JSP applications.
- **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 <http://javashoplm.sun.com/ECom/docs/Welcome.jsp?StoreId=8&PartDetailId=JWSDP-Plugin-1.6-G-F&TransactionId=try>.
- **Administrative Tools** – The Application Server includes a command-line tool and a browser-based Administration Console interface.
- **High Performance Message Delivery** – The Application supports concurrent message delivery with the Sun Java System Message Queue software.
- **Developer Tool Integration** – The Application Server supports the NetBeans IDE and Sun Java Studio Creator.
- **Admin Console GUI Enhancements** – The Admin Console's new capabilities include an improved log viewer, a JNDI namespace browser, and a GUI for monitoring.
- **Web Services Security** – These container message security mechanisms implement message-level authentication (e.g. 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** – This release implements Web Services Interoperability (WS-I) Basic Profile 1.1 to enable interoperability for web services applications.
- **Migration Tool** – The migration tool helps in moving applications running on other application servers such as JBoss, WebLogic and Websphere to this release of the Sun Java System Application Server.

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

Table 1 Major API changes on the J2EE 1.4 Platform (*continued*)

API	Description
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

Hardware and Software Requirements

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

- [Platform Requirements](#)
- [Important Patch Information](#)
- [JDBC Drivers and Databases](#)
- [Web Servers](#)
- [Browsers](#)
- [Upgrading the Sun Java System Application Server](#)
- [Other Requirements](#)

Platform Requirements

The following table lists the operating systems that are supported for Sun Java System Application Server Platform Edition 8.1 2005Q1 product.

Table 2 Supported Operating Systems

Operating System	Minimum Memory	Recommended Memory	Minimum Disk Space	Recommended Disk Space	JVM
Sun Solaris 8, 9, 10 (SPARC) Solaris 9, 10(x86)	256 MB	512 MB	250 MB free	500 MB free	J2SE 1.4.2_06
Sun Java Desktop System	256 MB	512 MB	250 MB free	500 MB free	J2SE 1.4.2_06
Redhat Enterprise Linux 2.1 Update 2, 3.0 Update 1	256 MB	512 MB	250 MB free	500 MB free	J2SE 1.4.2_06
Microsoft Windows 2000® Advanced Server Service Pack 4+	256 MB	512 MB	250 MB free	500 MB free	J2SE 1.4.2_06
Microsoft Windows XP® Professional Service Pack 1+	256 MB	512 MB	250 MB free	500 MB free	J2SE 1.4.2_06

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

Solaris Patch Requirements

It is recommended that Solaris 8, 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 Platform Edition 8.1 go to <http://sunsolve.sun.com> and select either “Patches” or “Patch Portal.” Follow the Sun Java System Application Server Platform 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 Platform Edition 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
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+
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 Platform Edition 8.1 2005Q1.

Table 5 Supported Web Servers

Web Server	Version	Operating System
Sun Java System Web Server	6.0 Service Pack 6+	Solaris SPARC 8 9, 10 Solaris x86 9, 10 Red Hat Enterprise Linux 2.1 Update 2, 3.0 Update 1

Table 5 Supported Web Servers

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

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

Browsers

This section lists the browsers that are supported with the Sun Java System Application Server Platform Edition 8.1 2005Q1.

Table 6 Browsers Supported

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

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 Platform Edition 8.1 2005Q1.

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 3820 and 3890 for IIOP/SSL). If these default port numbers are in use, the installation program will assign a random port number from the dynamic port range (note that this may not be the next available port number).
- **Starting previously-installed servers** (UNIX) — unless you are replacing the previously installed server, you should start it before you begin the Sun Java System Application Server 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>

Switching to J2SE 5.0

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 J2SE 5.0 instead of the bundled J2SE 1.4.2 to run the Application Server, perform the following steps (Windows and Unix):

1. Download the J2SE 5.0 SDK (not the JRE) and install it on your system, if you have not already done so.

The J2SE 5.0 SDK can be downloaded from <http://java.sun.com/j2se/1.5.0/>.

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 5.0 home directory:
4. Edit the *as-install/samples/common.properties* file, changing the line beginning “*com.sun.aas.javaRoot...*” to reference the J2SE 5.0 home directory.
5. Restart the Application Server.

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

NOTE If you are upgrading from a JDK version earlier than the bundled version (JDK 1.4.2_06) then you cannot upgrade to J2SE 5.0 using just the steps above. Specifically, in addition to the above steps, you must delete any existing domains and recreate them.

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>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>Administration Reference</i>	Editing the Sun Java System Application Server configuration file, <i>domain.xml</i> .

Table 7 Books in This Documentation Set (*continued*)

Book Title	Description
<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>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 Platform Edition 8.1 2005Q1 product. If a summary statement does not specify a particular platform, the problem applies to all platforms. This information is organized into the following sections:

- [Administration](#)
- [Application Client](#)
- [Container Managed Persistence](#)
- [Deploytool](#)
- [Documentation](#)
- [Installation](#)
- [Lifecycle Management](#)
- [Logging](#)
- [Security](#)
- [Upgrade Utility](#)
- [Web Container](#)

Administration

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

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

Solution

Do one of the following:

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

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*.

Cannot redeploy or undeploy the web module that is the default web module of any virtual server. (ID 6204799)

If the web module is specified as the default web module of a virtual server, and you try to redeploy or undeploy it, you will get the following error:

```
Trying to undeploy application from domain failed; Virtual Servers [server] have
<WEB-MODULE-NAME> as default web module. Please remove the default web module references
first. ; requested operation cannot be completed Virtual Servers [server] have
<WEB-MODULE-NAME> as default web module. Please remove the default web module references
first.
```

At this point, `domain.xml` is in an error state, and the Admin Console may not be able to display the table that shows the deployed web applications. The condition will persist even if the domain is stopped and started again.

Solution

1. Using the Admin Console, go to the virtual server page, and change the “default web module” to empty or specify another web module.
2. Using the CLI, undeploy the web module by specifying “domain” as the target.

```
# asadmin undeploy --target domain <WEB-MODULE-NAME>
```

The Admin Console should be fine now, and the web module can be deployed again, if desired.

FrameworkError exception after deploying a WAR and JAR to PE server via the AMX API in the Application Server GUI. (ID 6201462)

When an application is deployed on PE using the AMX API and not referenced, the Application Server GUI throws errors while displaying that application. AMX requires that you explicitly handle references for your applications. For example, when an application is deployed, the `DeployedItemRefConfig` needs to be explicitly created. To simplify the deployment process, references are assumed to be present in PE, which in turn causes the issue with Application Server GUI.

Solution

Always create the reference to a resource or application after creating it.

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.

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 `"jms/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.

Some documented monitoring features do not apply to Platform Edition. (ID 6202255)

The documentation for AMX (Application Server Management eXtensions) does not specify some monitoring features that are not available in Application Server Platform Edition 8.1. Specifically, the components that cannot be monitored in the Platform Edition are as follows:

- **Production Web Container (PWC):**

- PWC HTTP Service
- PWC Connection Queue
- PWC ThreadPool
- PWC DNS
- PWC KeepAlive
- PWC File Cache
- PWC Virtual Server
- PWC Request

- **Webmodule**

- SessionSize
- ContainerLatency
- SessionPersistTime
- CachedSessionsCurrent
- PassivatedSessionsCurrent

- **StatefulSessionStore**

- CheckpointCount
- CheckpointSuccessCount
- CheckpointErrorCount
- CheckpointedBeanSize
- CheckpointTime

Solution

None needed. These statistics are not relevant for Platform Edition.

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 Platform 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`.

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

- 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.

Installation

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

Intermittent failure to render “Next” navigation button on installer and uninstaller Welcome screen. (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/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:

```

Checking 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.

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-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.

Security

This section describes known security issues and solutions.

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.

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 Platform Edition 8 to Application Server Platform 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.

Port conflict when starting domain1 or samples domain after upgrading from 8.0 Platform Edition to 8.1 Platform Edition. (ID 6202188)

After upgrading an 8.0 Application Server with multiple domains, the domains may not be able to start simultaneously due to having the same port number configured for the JMX connector.

Solution

1. Check the *install_dir*/domains/domain1/config/domain.xml file, for the following entry:

```
<jmx-connector accept-all="false" address="0.0.0.0" auth-realm-name="admin-realm"
enabled="true" name="system" port="8686" protocol="rmi_jrmp"
security-enabled="false"/>" -- and in file <as 8.1 install
dir>/domains/domain1/samples/config/domain.xml, notice it used the same port "8686", so
it failed to start domain due to port conflict.
```

2. Change the port value 8686 to 8687, and then restart domain1.

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

```

Solution

Set the JSP compilation switch “fork” to “false.”

This can be done either of two ways:

- Globally, by setting the `fork init` parameter of the `JspServlet` in `${SIAS_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.

Performance degradation on multi-CPU machines. (ID 6194026)

The default configuration of the Application Server PE does not perform optimally on multi-CPU machines. A trade-off is made so that startup is faster, but this can negatively impact the performance of web applications.

Solution

Configure the Application Server to use the following JVM option:

```
-Dcom.sun.enterprise.server.ss.ASQuickStartup=false
```

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 Platform 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 Platform Edition 8.1 2005Q1 Release Notes*, and the part number is 819-0072.

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 intellectuels 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.

