Sun Java System Application Server Platform Edition 8.2 Release Notes

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

Part No: 819–4707–14
May 2008

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

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

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

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

080522@20209

Contents

1 **Overview** 7

About These Notes 7

Release Notes Revision History 8

Accessibility Features 8

Related Documentation 8

How to Report Problems and Provide Feedback 9

Sun Welcomes Your Comments 10

Additional Sun Resources 10

2 **About Application Server Platform Edition 8.2** 13

What's New in the 8.2 Release 13

Hardware and Software Requirements 14

- Platform Requirements 14
- System Virtualization Support 15
- Important Patch Information 15
- JDBC Drivers and Databases 16
- Using the Bundled Derby Database 17
- Browsers 20
- Upgrading the Sun Java System Application Server 20
- Other Requirements 21

Bugs Fixed in the 8.2 Release 21

J2EE Support 23

Switching to Another Supported J2SE Version 24

▼ To switch to another supported J2SE version 24

3 **Known Issues and Limitations** 27

Administration 27

The package-appclient script does not work if domain1 is not present. (ID 6171458) 27

Cannot restore backed-up domain with another name. (ID 6196993) 28

Starting Application Server with additional JMX Agent is not supported. (ID 6200011) ... 28

Cannot redeploy or undeploy the web module that is the default web module of any virtual server. (ID 6204799) 29

FrameworkError exception after deploying a WAR and JAR to PE server via the AMX API in the Application Server GUI. (ID 6201462) 29

Java Home Setting inside Configuration does not take effect. (ID 6240672) 30

Selector.select() throws IOException. App Server startup fails. (ID 6322825) 30

Domain fails to start when create-domain master password has special characters. (ID 6345947) 30

Specific Java System Properties are not handled correctly by AS 8.2 Startup. (ID 6372759) 31

Application Client 31

Library JAR packaged in Application Client Archive overwrites MANIFEST file. (ID 6193556) 31

Dynamic content technology such as CGI-bin and SHTML functionality not supported. (ID 6373043) 31

Database Driver 31

DB2 Server has connection growing after idle time-out with DB2 Type II driver (ID 2082209/5022904) 32

Deploytool 32

Deploytool often will not create message-destination elements in the following Sun deployment descriptors (ID 6197393): 32

“Home” incorrectly translated as “installation directory” in Deploytool for Simplified Chinese. (ID 6203658) 33

Documentation 33

Some documented monitoring features do not apply to Platform Edition. (ID 6202255) . 33

AppservPasswordLoginModule referenced as AbstractPasswordLoginModule in documentation (ID 6229682) 34

Incorrect -W short option for --passwordfile in 8.2 PE man pages. (ID 6373588) 34

The Javadoc for several AMX interfaces and methods is either missing or incorrect (several IDs): 35

Documentation on getting a physical Connection from a wrapped Connection is no longer correct (ID 6486123) 35

Installation 35

Intermittent failure to render the Next navigation button on installer and uninstaller Welcome screen. (ID 4977191) 35

Installation shutdown hanging on some Linux systems after clicking the Finish button. (5009728) ... 36

Intermittent J2SE detection and bootstrap issues in install wrapper on Linux. (6172980) ... 36

Application Server does not support NFS. ... 37

Lifecycle Management ... 37

After setting the `ejb-timer-service` property `minimum-delivery-interval` to `9000`, an attempt to set the `ejb-timer-service` property `redelivery-interval-in-mills` to `7000` causes the `set` command to fail with the following error: (ID 6193449) ... 37

Logging ... 38

Setting debug statement for `access.failure` causes hanging in Application Server startup. (ID 6180095) ... 38

Sample Applications ... 38

`managementws` sample needs to update `MANIFEST.MF` references from `castor-0.9.3.9-xml.jar` to `castor-0.9.9.1.jar`. (ID 6363339) ... 38

Security ... 40

WS security: appclient container is not properly integrated with JAXRPC client runtime. (ID 6325469) ... 40

Upgrade Utility ... 40

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 8.2. (ID 6165528) ... 41

Port conflict when starting `domain1` or `samples` domain after upgrading from 8.0 Platform Edition to 8.2 Platform Edition. (ID 6202188) ... 41

The installer running “Upgrade in place” fails to start upgrade tool on some Linux systems after clicking on the “Start Upgrade Wizard” button. (6207337) ... 41

Garbage characters displayed in Results panel after upgrade (ID 6376140) ... 42

Web Container ... 42

Deploying an application using `--precompilejsp=true` can lock JAR files in the application, causing later undeployment or redeployment to fail. (Windows only) (ID 5004315) ... 42

Unable to deploy WAR with Servlet 2.4-based `web.xml` that contains an empty `<load-on-startup\>` element. (ID 6172006) ... 43

Unable to compile JSP page on resource constrained servers. (ID 6184122) ... 44

Performance degradation on multi-CPU machines. (ID 6194026) ... 44

Received malformed Fast Infoset documents can disable Fast Infoset support for JAX-RPC deployed services. (ID 6368670) ... 45

CHAPTER 1

Overview

The Sun Java™ System **Application Server Platform Edition 8.2** 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 if free for development, deployment and redistribution. Customers interested in redistribution should contact Sun OEM sales (`http://www.sun.com/software/products/appsrvr/appsrvr_oem.html`) for a redistribution license.

This document contains the following sections:

- "About These Notes" on page 7
- "Release Notes Revision History" on page 8
- "Accessibility Features" on page 8
- "Related Documentation" on page 8
- "How to Report Problems and Provide Feedback" on page 9
- "Sun Welcomes Your Comments" on page 10
- "Additional Sun Resources" on page 10

About These Notes

These Release Notes contain important information available at the time of release of Sun Java System Application Server 8.2. New features and enhancements, known issues and limitations, and other information are addressed here. Read this document before you begin using Application Server 8.2.

The most up-to-date version of these release notes can be found at the Sun Java System web site (`http://docs.sun.com/app/docs/coll/1343.2`). Check the web site prior to installing and setting up your software and then periodically thereafter to view the most up-to-date release notes and product documentation.

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

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

Release Notes Revision History

This section lists the revision history of these release notes.

TABLE 1–1 Release Notes Revision History

Revision Date	Description
January 2006	Initial release of the Sun Java SystemApplication Server 8.2 product.
February 2006	General editorial fixes, updated URL to Red Hat patch RPM.
March 2006	Additional issues documented, list of fixed bugs updated.
July 2007	Added defect 6396045 to known problems for installation.
May 2008	Added System Virtualization Support section.

Accessibility Features

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

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

Related Documentation

In addition to these release notes, the Application Server product includes an entire documentation set (`http://docs.sun.com/app/docs/coll/1343.2`).

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

TABLE 1–2 Books in This Documentation Set

Book Title	Description
Sun Java System Application Server Platform Edition 8.2 Quick Start Guide	How to get started with the Sun Java System Application Server product.
Sun Java System Application Server Platform Edition 8.2 Installation Guide	Installing the Sun Java System Application Server software and its components.
Sun Java System Application Server Platform Edition 8.2 Developer's Guide	Creating and implementing Java^{TM} 2 Platform, Enterprise Edition (J2EE^{TM} 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.
Sun Java System Application Server Platform Edition 8.2 J2EE Tutorial	Using J2EE 1.4 platform technologies and APIs to develop J2EE applications and deploying the applications on the Sun Java System Application Server.
Sun Java System Application Server Platform Edition 8.2 Administration Guide	Configuring, managing, and deploying the Sun Java System Application Server subsystems and components from the Administration Console.
Sun Java System Application Server Platform Edition 8.2 Administration Reference	Editing the Sun Java System Application Server configuration file, `domain.xml`.
Sun Java System Application Server Platform Edition 8.2 Upgrade and Migration Guide	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.
Sun Java System Application Server Platform Edition 8.2 Troubleshooting Guide	Solving Sun Java System Application Server problems.
Sun Java System Application Server Platform Edition 8.2 Error Message Reference	Solving Sun Java System Application Server error messages.
Sun Java System Application Server Platform Edition 8.2 Reference Manual	Utility commands available with the Sun Java System Application Server; written in manpage style. Includes the `asadmin` command line interface.

How to Report Problems and Provide Feedback

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

- **Feedback Submittal form**– A form (`http://java.sun.com/docs/forms/J2EE14SubmittalForm.html`) for submitting feedback on the Application Server product.
- **J2EE-INTEREST list**– A mailing list (`http://archives.java.sun.com/archives/j2ee-interest.html`) for J2EE questions.
- **Bug database on Java Developer Connection**– To view bugs or to submit a bug, use the Java Developer Connection Bug Parade (`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 (`http://forum.java.sun.com/`) for discussions related to the Sun Java System **Application Server Platform Edition 8.2** product.
- **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 8.2 Release Notes*, and the part number is 819-4707.

Additional Sun Resources

Useful information can be found at the following locations:

- Application Server product information (`http://wwws.sun.com/software/products/appsrvr/home_appsrvr.html`)
- Application Server product documentation (`http://docs.sun.com/app/docs/coll/1343.2`)
- 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://wwws.sun.com/software`)
- Sun Microsystems product documentation (`http://docs.sun.com/`)

CHAPTER 2

About Application Server Platform Edition 8.2

The Sun Java™ System **Application Server Platform Edition 8.2** is a J2EE 1.4 platform-compatible server for the development and deployment of J2EE applications and Java technology-based web services.

This chapter includes:

- "What's New in the 8.2 Release" on page 13
- "Hardware and Software Requirements" on page 14
- "Bugs Fixed in the 8.2 Release" on page 21
- "J2EE Support" on page 23
- "Switching to Another Supported J2SE Version" on page 24

What's New in the 8.2 Release

The Sun Java System Application Server Platform Edition 8.2 implements many new features:

- J2EE 1.4 compatible.
- *(Improved)* Outstanding developer experience with NetBeans 5 and improved deployment speed, runtime footprint and server start up time. NetBeans 5.0 includes Application Server 8.2 as a default J2EE runtime.
- *(New)* Superior throughput performance and better scalability supporting multi core/multi threaded architecture.
- *(New)* Fast Infoset support improves web services performance multiple folds.
- *(New)* Application Server 8.2 adds support for Red Hat 4 and My SQL 5. See "Hardware and Software Requirements" on page 14, later in these notes, for a complete list of supported operating systems and database drivers.
- *(New)* Application Server 8.2 bundles deployment-ready Derby database, making it possible to develop and deploy end-to-end J2EE applications.
- *(New)* Built-in JMS resource adapter makes connectivity with backend systems even easier. Application Server 8.2 supports connectivity to IBM MQ Series and Sun's MQ Server.

- *(Updated)* Enhanced J2EE 1.4 blueprints including demonstration of using Web 2.0 technologies like AJAX with Application Server 8.2.
- JavaServer Faces Support — Developers can quickly build web applications by assembling reusable UI components in a page, connecting these components to a data source, and wiring client-generated events to server-side event handlers.
- Improved Administration Console — The Application Server 8.2 Administration Console adds new look and feel with capabilities like JNDI namespace browser, JDBC Connection Validation, Deployment Descriptor Viewer, Error log viewer and improved monitoring.
- Web Services Security: Container message security mechanisms implementing 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.
- JavaServer Pages Standard Tag Library 1.1 Support: the library encapsulates core functionality common to many JSP applications.
- *(Updated)* Bundled J2SE 5.0_06.

The Sun Java System Application Server Platform Edition is free for development, deployment and redistribution. Customers interested in redistribution should contact Sun OEM (`http://www.sun.com/software/products/appsrvr/appsrvr_oem.html`) sales for a redistribution license. Sun provides support at additional cost. If you would like to be contacted about licensing the Java 2 Platform, Enterprise Edition, fill out this form (`http://java.sun.com/j2ee/license_form.html`).

Hardware and Software Requirements

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

- "Platform Requirements" on page 14
- "System Virtualization Support" on page 15
- "Important Patch Information" on page 15
- "JDBC Drivers and Databases" on page 16
- "Using the Bundled Derby Database" on page 17
- "Browsers" on page 20
- "Upgrading the Sun Java System Application Server" on page 20
- "Other Requirements" on page 21

Platform Requirements

The following table lists the operating systems that are supported for Sun Java System Application Server Platform Edition 8.2 product.

TABLE 2–1 Supported Operating Systems

Operating System	Minimum Memory	Recommended Memory	Minimum Disk Space	Recommended Disk Space	JVM
Sun Solaris 9, 10 (SPARC) Solaris 9, 10(x86)	512 MB	512 MB	250 MB free	500 MB free	J2SE 1.4.2_10, J2SE 5_06
Sun Java Desktop System	512 MB	1 GB	250 MB free	500 MB free	J2SE 1.4.2_10, J2SE 5_06
Redhat Enterprise Linux 3.0 U1, 4.0	512 MB	1 GB	250 MB free	500 MB free	J2SE 1.4.2_10, J2SE 5_06
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_10, J2SE 5_06

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

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

System Virtualization Support

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

Important Patch Information

For the current list of required patches for Sun Java System Application Server Platform Edition 8.2 go to `http://sunsolve.sun.com` and select either "Patches" or "Patch Portal." Follow the Sun Java System Application Server Platform Edition 8.2 links. As operating system patch requirements change and patches to Java Enterprise System components become available, updates will be made available on SunSolve, initially in the form of recommended patch clusters.

Solaris Patch Requirements

It is recommended that Solaris 8, 9, 10 (x86, SPARC) users have the “Sun recommended patch cluster” installed. This patch cluster is available under “Recommended and Security Patches” on the SunSolve (`http://sunsolve.sun.com/`) web site.

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

JDBC Drivers and Databases

The Sun Java System Application Server 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 2–2 J2EE-Compatible JDBC Drivers

JDBC Vendor	JDBC Driver Type	Supported Database Server
i-net Software	Type 4	Oracle (R) 8.1.7, 9i, 9.2.0.3+, 10.1.*x*, 10.2.*x* Sybase ASE 12.5. Microsoft SQL Server 2000 4.0 Service Pack 1
IBM	Type 2	IBM DB2 8.1 Service Pack 3+
Derby	Type 4	Apache Derby 10.1.2.1
PointBase	Type 4	PointBase Network Server 5.2
MySQL	Type 4	5.*x*
DataDirect	Type 4	Oracle (R) 8.1.7, 9i, 9.2.0.3+, 10.1.*x*, 10.2.*x* Sybase ASE 12.5.2 Microsoft SQL Server IBM DB2 8.1 Service Pack 3+
Oracle	Type 4, Type 2	Oracle (R) 9.2.0.3+, 10.1.*x*, 10.2.*x*

Using the Bundled Derby Database

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

- "Starting and Stopping the Derby Database" on page 17
- "Derby Utility Scripts" on page 17
- "Exporting Tables from Pointbase to Derby" on page 18

Starting and Stopping the Derby Database

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

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

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

  The default value for the host is `0.0.0.0`, which allows for Derby to listen on `localhost` as well as the IP/hostname interfaces. The value for the `dbhome` property represents the location of where the Derby databases reside. The default `path` is *<appserver_install_dir>*`/derby`.
- The `asadmin stop-database` command is used to shut down an instance of the Derby network server that is running:

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

Derby Utility Scripts

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

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

▼ To Configure Your Environment to Run the Derby Utility Scripts

1. **Set the `DERBY_INSTALL` environment variable to point to the** *<appserver_install_dir>*`/derby` **directory.**

2. **Unset your `CLASSPATH` environment variable.**

3. **You can also optionally set the following properties:**

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

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

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

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

Exporting Tables from Pointbase to Derby

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

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

PointBase Commander 5.2 ECF build 294 size restricted version EMBEDDED

Interactive SQL command language. SunOS/5.9

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

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

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

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

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

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

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

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

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

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

Browsers

This section lists the browsers that are supported with the Sun Java System Application Server Platform Edition 8.2 administration console and Quick Start Guide. The browsers supported when running applications on the Application Server depend on the applications being run.

TABLE 2–3 Browsers Supported

Browser	Version
Mozilla	1.4, 1.5, 1.6, 1.7.*x*
Netscape Navigator	6.2, 7.0
Internet Explorer	5.5 Service Pack 2, 6.0
Firefox	1.*x*

Upgrading the Sun Java System Application Server

Refer to the *Sun Java System Application Server Platform Edition 8.2 Installation Guide* for complete instructions for upgrading from a previous version of the Application Server to the Sun Java System Application Server Platform Edition 8.2.

Other Requirements

The following additional requirements should be met before installing the Sun Java System Application Server software.

- **Free space** — Your temporary directory must have a minimum of 200 MB free for Sun Java System Application Server installation, and 250 MB of free space for the SDK installation.
- **Using the uninstall program**– If you need to remove the Application Server from your system, it is important to use the uninstall program that is included with the software. If you attempt to use another method, problems will arise when you try to reinstall the same version, or when you install a new version.
- **Free ports**– You must have seven unused ports available.
  - The installation program automatically detects ports in use and suggests currently unused ports for the default settings. By default, the initial default ports are `8080` for HTTP, `8181` for HTTPS, and `4848` for the Administration Server.
  - The installation program will detect used ports and assign two others for you: Sun JavaTM System Message Queue (by default, `7676`), and IIOP (by default, `3700` for IIOP and `3820` and `3890` for IIOP/SSL). If these default port numbers are in use, the installation program will assign a random port number from the dynamic port range (note that this may not be the next available port number).

  Starting previously-installed servers (UNIX) – Unless you are replacing the previously installed server, you should start it before you begin the Sun Java System Application Server 8.2 installation process. This allows the installation program to detect ports that are in use and avoid assigning them for other uses.
- **Replacing previously-installed servers** (UNIX) – If you have an older version on the Sun Java System Application Server installed that you wish to replace with the current Application Server, you should stop it before installing the new server.
- **Shutting down firewall** (Microsoft Windows) – You must stop any firewall software before installing the Sun Java System Application Server software, because some of this software disables all ports by default. The installation program must be able to accurately determine which ports are available.

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

Bugs Fixed in the 8.2 Release

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

6198981 Broken panels in the New Web Service Wizard.

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

6155080	Specifying target message by `java-method` does not work in `client-side message-security-binding` elements.
6173248	Using the AMX API, removing a J2EE application reference from a server removes the application, but the application is still accessible.
6360646	AS 8.2 PE/EE: Use the `PlatformMBeanServer` if available.
6295215	`DOCS:java.sql.statement.getConnection()` does not meet javadoc specification for pooling connections.
6290666	Port `8080` is not released by domain even after stopping and deleting.
6286688	Cannot save transaction support value when JMS connection factory was created for the first time.
6298257	For Application Server 8.1 UR2 PE, `asant` cannot be run on Windows 2000 at all; command too long error is returned.
6320008	Rich client RMI-IIOP failover testing fails.
6347544	Bundle Grizzly ARA supports in 8.2.
6275566	Application Server 8.1 Virtual Server access log location not updating.
6351023	Generic RA for JMS need to be integrated with AS 8.2.
6288752	Disk I/O for XA transaction logs too high.
6356910	Make Application Server 8.x native launcher Bourne shell-friendly by disassociating from controlling TTY.
6307510	S1AS 7.0/SJAS7.1 : EJBC/RMIC generates STUB/Skel with NOT fully Qualified Package Name.
6286783	Server has to reject requests with double `Content-Length` header
6207862	`asadmin create-domain --help` produces some CVS merge characters and is garbled.
6377830	`setAutoCommit` to `false` gets propagated when the same connection is used by the next user.
6317857	Undeploy: Error unregistering mbean.
6284124	Servlet container UTF-8 URI mapping issue.
6276218	Deploytool does not work with spaces in the install path
6211979	Deploy command fails on file based non-root installation.
6354545	Deployment hangs in Windows.
6283805	Deployed applications cannot be accessed after upgrade.

6327037	Deployment performance improvement needed.
6270387	Redeploy sometimes fails with `Error while running ejbc -- Fatal Error from EJB Compiler`.
6258619	Undeployment does not release all files.
6276021	Redeployment of WAR file (remote deployment for Creator) fails.
6330332	AS8102 memory leaks on deploy/undeploy scenario (SubCR from 6324399 `EJBClassLoader`.

J2EE Support

The Sun Java System Application Server 8.2 supports the J2EE 1.4 and Java EE 5 platforms. The following table describes the enhanced APIs available on the J2EE 1.4 platform.

TABLE 2–4 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	

TABLE 2–4 Major API changes on the J2EE 1.4 Platform (Continued)

API	Description
J2EE Deployment 1.1	Standard APIs that enable deployments of J2EE components and applications
J2EE Management 1.0	Definitions for the information model for managing the J2EE platform
Java Management Extensions (JMX) 1.2	Standard management API
Java Authorization Contract for Containers (JACC) 1.0	Definitions of security contracts between a J2EE Application Server and the authorization policy provider
Java API for XML Processing (JAXP) 1.2	An API with which applications can parse and transform XML documents; also adds support for processing of XML schemas
JMS 1.1	A messaging standard that enables J2EE application components to create, send, receive, and read messages; also adds support for uniform APIs for queues and topics
JavaMail 1.3	A set of abstract classes that model a mail system; also includes minor updates to the APIs

Switching to Another Supported J2SE Version

Sun Java System Application Server 8.2 supports both J2SE 1.4.2 and J2SE 5.0 as the underlying JVM. If you want to switch from one J2SE version to another, perform the following general steps. (Windows and Unix)

▼ To switch to another supported J2SE version

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

The J2SE SDK can be downloaded from `http://java.sun.com/j2se`.

2 **Completely stop the Application Server.**

You can use the following command line:

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

Alternatively, you can use the Administration Console GUI:

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

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

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

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

5 **Restart the Application Server.**

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

Next Steps 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 or later using just the steps above. Specifically, in addition to the above steps, you must delete any existing domains and recreate them.

CHAPTER 3

Known Issues and Limitations

This chapter describes known problems and associated workarounds for the Sun Java System Application Server Platform Edition 8.2 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" on page 27
- "Application Client" on page 31
- "Database Driver" on page 31
- "Deploytool" on page 32
- "Documentation" on page 33
- "Installation" on page 35
- "Lifecycle Management" on page 37
- "Logging" on page 38
- "Sample Applications" on page 38
- "Security" on page 40
- "Upgrade Utility" on page 40
- "Web Container" on page 42

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 8.2 work only for backing up and restoring the same domain on the same machine.

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

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

Example values include:

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

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

Solution

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

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

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

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

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

Change the default web module.

▼ To change the default web module

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.

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

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

Solution

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

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

`Selector.select()` throws `IOException`. App Server startup fails. (ID 6322825)

In the current JDK code, the `/dev/poll` Selector allocates an array of `8192` `pollfd` entries for use by the Selector. This is exceeds the `nofiles` `ulimit`, causing it to fail with an “invalid argument” error. This in turn causes the App Server socket service that connects to MQ during startup to fail with an `IOException` because the `selector.select()` is broken.

Solution

Increase the `pollfd` file descriptor limit. There are two ways to do this:

1. Execute `ulimit -n 8193` on the shell as root.
2. Increase the hard limit on the number of file descriptors to `8193` or higher:
   a. Check the hard limit with `ulimit -n -H`.
   b. If less than `8193`, edit `/etc/system`, adding the `set rlim_fd_max=8193` command.
   c. Reboot the machine.

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

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

Solution

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

Specific Java System Properties are not handled correctly by AS 8.2 Startup. (ID 6372759)

Adding the following to the JVM proxy settings causes the server to not start:

```
<jvm–options>-Dhttp.proxyHost=webcache.east.sun.com</jvm–options>
<jvm–options> -Dhttp.proxyPort=8080</jvm–options>
<jvm–options>-Dhttp.nonProxyHosts="mssp.ctu.gov|*.ctu.gov|localhost"
</jvm–options>
```

Inserting a `*` character causes a No Class Def Found error (Exception in thread `main java.lang.NoClassDefFoundError: com/sun/enterprise/security/store/IdentityManager`). Inserting a `|` character causes the start script to timeout waiting for server to start.

This functionality is critical in order to support Application Server Deployments (and Portal deployments) that reside behind a firewall and need access to both external and internal servers. An example is the Portal Server URL Scraper. These settings are necessary in order to allow the URL Scraper to get content from external sources.

Solution

Edit `install-dir/config/asenv.conf` file, changing the line `AS_NATIVE_LAUNCHER="true"` to `AS_NATIVE_LAUNCHER="false"`.

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.

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

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

Solution

Use JSP and Web service technologies instead.

Database Driver

This section describes the known database driver issues and associated solutions.

DB2 Server has connection growing after idle time-out with DB2 Type II driver (ID 2082209/5022904)

After porting applications from another application server, physical connections are not being closed properly after the connections timing out. This problem is seen with the DB2 8.1.x version of the Client libraries (Type II) Driver against the same DB2 7.1.x Database Server.

Solution

Set the `SteadyPoolSize` and `MaxPoolSize` to the same number, and in addition, set the `Idle Connection` timeout also to `0` (zero). This disables the timing-out of idle connections and the user will have the full set of connections available.

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:

To edit an existing JNDI name

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:

5 **Restart deploytool.**

6 **On the Message Destinations tab, select a Message Destination or add a new Message Destination.**

7 **Enter the JNDI Name for the Message Destination in the Sun-specific JNDI Name text field, and then press Enter.**

8 **Review the Sun descriptor by clicking Tools\>Descriptor Viewer\>Application Server Descriptor.**

9 **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.

“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 eXtenstions) does not specify some monitoring features that are not available in Application Server Platform Edition 8.2. 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.

`AppservPasswordLoginModule` referenced as `AbstractPasswordLoginModule` in documentation (ID 6229682)

The "Realm Configuration" in *Sun Java System Application Server Platform Edition 8.2 Developer's Guide* section in Chapter Chapter 2, "Securing Applications," in *Sun Java System Application Server Platform Edition 8.2 Developer's Guide* in the*Sun Java System Application Server Platform Edition 8.2 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`.

Incorrect `-W` short option for `--passwordfile` in 8.2 PE man pages. (ID 6373588)

There should be no short option for `--passwordfile`. Currently `-W --passwordfile` is documented in the man pages. This is incorrect.

Solution

Do not attempt to use the –W option with --passwordfile with Application Server 8.2 Platform Edition. The short option is scheduled to be added to a future Application Server release.

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.

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

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

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

should now read:

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

Installation

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

Intermittent failure to render the Next navigation button on installer and uninstaller Welcome screen. (ID 4977191)

This problem has been reported intermittently on the Solaris *x*86 platform, but it is possible that it also affects Solaris SPARC and Linux platforms.

The problem is that the installer\qs or uninstaller\qs 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:

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

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

Solutions

To work around this issue:

▼ To work around the bootstrap issues on Linux

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

Application Server does not support NFS.

This versions of Application Server does not support Network File System (NFS).

Solution

None.

Lifecycle Management

This section describes known lifecycle management issues and associated solutions.

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

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

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

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

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

Solution

Use the default values for these properties, as follows:

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

Values other than these defaults will generate an error.

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.

Sample Applications

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

`managementws` sample needs to update `MANIFEST.MF` references from `castor-0.9.3.9-xml.jar` to `castor-0.9.9.1.jar`. (ID 6363339)

When running the verifier on `<install_dir>/samples/webservices/jaxrpc/apps/managementws`, you encounter the following warnings:

```
[exec] WARNING: /var/tmp/exploded20051214111425/managementws/ \
managementwsEjb_jar contains library/castor-0.9.3.9-xml.jar in Class-Path
manifest attribute, but it is not found in ear file
[exec] Dec 14, 2005 11:14:30 AM Archive getBundledArchives
[exec] WARNING: /var/tmp/exploded20051214111425/managementws/ \
managementwsEjb_jar contains library/castor-0.9.3.9-xml.jar in Class-Path
manifest attribute, but it is not found in ear file
```

The Castor jar was updated in the Application Server 8.2 release, so all references to the older `castor-0.9.3.9-xml.jar` should be changed to point to the newer `castor-0.9.9.1.jar`. Specifically you need to change references in the `MANIFEST.MF` files to use `castor-0.9.9.1.jar` instead of the older `castor-0.9.3.9-xml.jar`.

Solution

Change the following references to the older Castor jar to point to the new Castor jar:

Old:

```
src/conf/MANIFEST.MF:Class-Path: library/castor-0.9.3.9-xml.jar
src/conf/MANIFEST.MF:Name: library/castor-0.9.3.9-xml.jar
managementws-ejb/src/conf/MANIFEST.MF:Class-Path: \
library/castor-0.9.3.9-xml.jar
```

New:

```
src/conf/MANIFEST.MF:Class-Path: library/castor-0.9.9.1.jar
src/conf/MANIFEST.MF:Name: library/castor-0.9.9.1.jar
managementws-ejb/src/conf/MANIFEST.MF:Class-Path: \
library/castor-0.9.9.1.jar
```

Next, clean up the `build.xml` file so it does not copy the Castor `.jar` to `install_dir/lib` during deployment and remove it during undeployment. The following are diffs of the old and new `build.xml` files.

```
% cvs diff build.xml Index: build.xml
===================================================================
RCS file: /m/jws/samples/samples8x/webservices/jaxrpc/apps/managementws/ \
managementws-standalone-client/ Attic/build.xml,v retrieving revision \
1.1.2.3
diff -r1.1.2.3 build.xml
80,89d79
<   <target name="remove_castor_from_classpath">
<       <delete file="${com.sun.aas.installRoot}/lib/castor-0.9.9.1.jar"/>
<   </target>
<   <target name="add_castor_to_classpath">
<       <delete file="${com.sun.aas.installRoot}/lib/castor-0.9.9.1.jar"/>
<       <copy file="../lib/castor-0.9.9.1.jar" \
            todir="${com.sun.aas.installRoot}/lib" />
<   </target>
<
<   <target name="setup" depends="add_castor_to_classpath, restart.server"/>
<  jbenoit/galapago 196 >pwd
/net/galapago.east/files/share/8.2ws/samples/samples8x/webservices/jaxrpc \
/apps/managementws/managementws-standalone-client
jbenoit/galapago 197 >cd ..
jbenoit/galapago 198 >cvs diff build.xml
Index: build.xml
===================================================================
RCS file: /m/jws/samples/samples8x/webservices/jaxrpc/apps/managementws/ \
Attic/build.xml
```

```
v retrieving revision 1.1.2.4
diff -r1.1.2.4 build.xml
28,36d27
<   <target name="setup">
<       <ant antfile="build.xml" inheritAll="true" dir="${sample.name}$ \
{standalone-client-dir-suffix}" target="setup"/>
<   </target>
<
<   <target name="unsetup">
<       <ant antfile="build.xml" inheritAll="true" dir="${sample.name}$ \
{standalone-client-dir-suffix}" target="remove_castor_from_classpath"/>
<   </target>
<
<
53,54c44,45
<   <target name="deploy"   depends="select_binary_common, deploy_common,
    setup" />
<   <target name="undeploy" depends="init,  undeploy_common, unsetup"/>
---
>   <target name="deploy"   depends="select_binary_common, deploy_common" />
>   <target name="undeploy" depends="init,  undeploy_common"/>
```

Security

This section describes known security issues and solutions.

WS security: appclient container is not properly integrated with JAXRPC client runtime. (ID 6325469)

The application client does not pass the user name and password to another Web Service client.

Solution

Pass the user name/password combination, if required, explicitly to the client program, as follows:

```
((Stub)yourWSPort)._setProperty(Stub.USERNAME_PROPERTY, "yourUsername");
((Stub)yourWSPort)._setProperty(Stub.PASSWORD_PROPERTY, "yourPassword");
```

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 8.2. (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 8.2 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

Change the port value.

▼ To change the port value

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.

▼ To use command line installation mode

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

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

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

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

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

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

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

Garbage characters displayed in Results panel after upgrade (ID 6376140)

When upgrading from the multilanguage version of Application Server 8.2 to a later version using some locales, the Results panel may display garbage characters, and the `/opt/SUNWappserver/domains/upgrade.log` file may also display garbage characters.

Solution

None at this time. This problem will be fixed in a future Application Server release.

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\qs` directory and the locked jar files remain on the server. The `server\qs` log file will contain messages describing the failure to delete the files and the `application\qs` 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\qs 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\qt 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\qs` other servlets. An empty `<load-on-startup\>` indicates that the order is irrelevant, as long as the servlet is loaded and initialized during the startup of its containing web application.

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

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

Solution

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

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

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

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

Solution

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

This can be done either of two ways:

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

  ```
  <servlet\> <servlet-name\>jsp</servlet-name\> <servlet-class\>org.apache.
  jasper.servlet.JspServlet</servlet-class\> .... <init-param\> <param-name\>
  fork</param-name\> <param-value\>false</param-value\> </init-param\> ....
  </servlet\>
  ```
- On a per-web application basis, by setting the `fork` JSP configuration property in `sun-web.xml` to `false`:

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

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

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
```

Received malformed Fast Infoset documents can disable Fast Infoset support for JAX-RPC deployed services. (ID 6368670)

If a nonconformant Fast Infoset encoded SOAP message is sent to a JAX-RPC service then the service will correctly fault in response. However, subsequent conformant Fast Infoset encoded SOAP messages sent to the same service or a service deployed using the same JAX-RPC runtime may fault incorrectly.

Solution

The following workarounds are possible:

- Disable Fast Infoset support on clients so that only XML encoded SOAP messages are sent.
- Restart the container deploying the services so that conforming Fast Infoset encoded SOAP messages can be sent.