



Sun Java System Application Server Platform Edition 8.2 Troubleshooting Guide

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

Part No: 819-4726
2005Q2

Copyright 2005 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. All rights reserved.

This product or document is protected by copyright and distributed under licenses restricting its use, copying, distribution, and decompilation. No part of this product or document may be reproduced in any form by any means without prior written authorization of Sun and its licensors, if any. Third-party software, including font technology, is copyrighted and licensed from Sun suppliers.

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

Sun, Sun Microsystems, the Sun logo, docs.sun.com, AnswerBook, AnswerBook2, and Solaris are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

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

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.

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

Copyright 2005 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. Tous droits réservés.

Ce produit ou document est protégé par un copyright et distribué avec des licences qui en restreignent l'utilisation, la copie, la distribution, et la décompilation. Aucune partie de ce produit ou document ne peut être reproduite sous aucune forme, par quelque moyen que ce soit, sans l'autorisation préalable et écrite de Sun et de ses bailleurs de licence, s'il y en a. Le logiciel détenu par des tiers, et qui comprend la technologie relative aux polices de caractères, est protégé par un copyright et licencié par des fournisseurs de Sun.

Des parties de ce produit pourront être dérivées du système Berkeley BSD licenciés par l'Université de Californie. UNIX est une marque déposée aux États-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company, Ltd.

Sun, Sun Microsystems, le logo Sun, docs.sun.com, AnswerBook, AnswerBook2, et Solaris sont des marques de fabrique ou des marques déposées, de Sun Microsystems, Inc. aux États-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 États-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

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

CETTE PUBLICATION EST FOURNIE "EN L'ETAT" ET AUCUNE GARANTIE, EXPRESSE OU IMPLICITE, N'EST ACCORDEE, Y COMPRIS DES GARANTIES CONCERNANT LA VALEUR MARCHANDE, L'APTITUDE DE LA PUBLICATION A REpondre A UNE UTILISATION PARTICULIERE, OU LE FAIT QU'ELLE NE SOIT PAS CONTREFAISANTE DE PRODUIT DE TIERS. CE DENI DE GARANTIE NE S'APPLIQUERAIT PAS, DANS LA MESURE OU IL SERAIT TENU JURIDIQUEMENT NUL ET NON AVENU.



060127@13215



Preface

The *Sun Java System Application Server Platform Edition 8.2 Troubleshooting Guide* describes common problems encountered when using the Sun Java System Application Server Server 8.2 Enterprise Edition and Platform Edition, and how to solve them.

Who Should Use This Book

The *Sun Java System Application Server Platform Edition 8.2 Troubleshooting Guide* is intended for developers and administrators who use the Sun Java System Application Server to assemble and deploy distributed and Web-based applications.

- Java APIs as defined in the JavaTM ServletTM, JavaServer PagesTM (JSPTM), Enterprise JavaBeansTM (EJBTM), and JavaTM Database Connectivity (JDBCTM) specifications
- The SQL structured database query languages
- Relational database concepts

Before You Read This Book

[If the reader must be familiar with other books before reading this book, list those books here. This section is not intended for *recommended* reading. You include that information later in the section titled "Related Books."]

[If the reader must know how to do something or must have completed some task before using this book, provide that information here.]

Application Server can be purchased by itself or as a component of Sun Java™ Enterprise System (Java ES), a software infrastructure that supports enterprise applications distributed across a network or Internet environment. If you purchased Application Server as a component of Java ES, you should be familiar with the system documentation at <http://docs.sun.com/coll/1286.1>.

How This Book Is Organized

This book describes workarounds for some problems you may encounter when configuring, using, or deploying applications with the Sun Java System Application Server 8.2 Enterprise Edition or Platform Edition software. The following table summarizes the content of this book.

- [Chapter 1](#) provides a general overview of the troubleshooting process.
- [Chapter 2](#) covers the most common problems you may encounter when using the product.
- [Chapter 3](#) covers problems related to security settings.
- [Chapter 4](#) covers commonly asked questions about Application Server software.

Application Server Documentation Set

The Application Server documentation set describes deployment planning and system installation. The URL for Application Server documentation is <http://docs.sun.com/coll/unknown>. For an introduction to Application Server, refer to the books in the order in which they are listed in the following table.

TABLE P-1 Books in the Application Server Documentation Set

Book Title	Description
<i>Release Notes</i>	Late-breaking information about the software and the documentation. Includes a comprehensive, table-based summary of the supported hardware, operating system, JDK, and JDBC/RDBMS.
<i>Quick Start Guide</i>	How to get started with the Application Server product.
<i>Installation Guide</i>	Installing the software and its components.

TABLE P-1 Books in the Application Server Documentation Set (Continued)

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

Related Books

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

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

The URL for all documentation about Java ES and its components is <http://docs.sun.com/prod/ent.sys.05q4>.

Default Paths and File Names

The following table describes the default paths and file names that are used in this book.

TABLE P-2 Default Paths and File Names

Placeholder	Description	Default Value
<i>install-dir</i>	Represents the base installation directory for Application Server.	<p>Sun Java Enterprise System installations on the Solaris™ platform:</p> <p><code>/opt/SUNWappserver/appserver</code></p> <p>Sun Java Enterprise System installations on the Linux platform:</p> <p><code>/opt/sun/appserver/</code></p> <p>Other Solaris and Linux installations, non-root user:</p> <p><code>user's home directory/SUNWappserver</code></p> <p>Other Solaris and Linux installations, root user:</p> <p><code>/opt/SUNWappserver</code></p> <p>Windows, all installations:</p> <p><code>SystemDrive:\Sun\AppServer</code></p>
<i>domain-root-dir</i>	Represents the directory containing all domains.	<p>Sun Java Enterprise System installations on the Solaris platform:</p> <p><code>/var/opt/SUNWappserver/domains/</code></p> <p>Sun Java Enterprise System installations on the Linux platform:</p> <p><code>/var/opt/sun/appserver/domains/</code></p> <p>All other installations:</p> <p><code>install-dir/domains/</code></p>
<i>domain-dir</i>	<p>Represents the directory for a domain.</p> <p>In configuration files, you might see <i>domain-dir</i> represented as follows:</p> <p><code>\${com.sun.aas.instanceRoot}</code></p>	<code>domain-root-dir/domain-dir</code>

TABLE P-2 Default Paths and File Names (Continued)

Placeholder	Description	Default Value
<i>instance-dir</i>	Represents the directory for a server instance.	<i>domain-dir/instance-dir</i>

Typographic Conventions

The following table describes the typographic changes that are used in this book.

TABLE P-3 Typographic Conventions

Typeface	Meaning	Example
AaBbCc123	The names of commands, files, and directories, and onscreen computer output	Edit your <code>.login</code> file. Use <code>ls -a</code> to list all files. <code>machine_name% you have mail.</code>
AaBbCc123	What you type, contrasted with onscreen computer output	<code>machine_name% su</code> Password:
<i>AaBbCc123</i>	A placeholder to be replaced with a real name or value	The command to remove a file is <code>rm filename</code> .
<i>AaBbCc123</i>	Book titles, new terms, and terms to be emphasized (note that some emphasized items appear bold online)	Read Chapter 6 in the <i>User's Guide</i> . <i>A cache</i> is a copy that is stored locally. Do <i>not</i> save the file.

Symbol Conventions

The following table explains symbols that might be used in this book.

TABLE P-4 Symbol Conventions

Symbol	Description	Example	Meaning
[]	Contains optional arguments and command options.	ls [-l]	The -l option is not required.
{ }	Contains a set of choices for a required command option.	-d {y n}	The -d option requires that you use either the y argument or the n argument.
\${ }	Indicates a variable reference.	\${com.sun.javaRoot}	References the value of the com.sun.javaRoot variable.
-	Joins simultaneous multiple keystrokes.	Control-A	Press the Control key while you press the A key.
+	Joins consecutive multiple keystrokes.	Ctrl+A+N	Press the Control key, release it, and then press the subsequent keys.
→	Indicates menu item selection in a graphical user interface.	File → New → Templates	From the File menu, choose New. From the New submenu, choose Templates.

Accessing Sun Resources Online

The docs.sun.comSM web site enables you to access Sun technical documentation online. You can browse the docs.sun.com archive or search for a specific book title or subject. Books are available as online files in PDF and HTML formats. Both formats are readable by assistive technologies for users with disabilities.

To access the following Sun resources, go to <http://www.sun.com>:

- Downloads of Sun products
- Services and solutions
- Support (including patches and updates)
- Training
- Research
- Communities (for example, Sun Developer Network)

Third-Party Web Site References

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 or alleged to be caused by or in connection with use of or reliance on any such content, goods, or services that are available on or through such sites or resources.

Sun Welcomes Your Comments

Sun is interested in improving its documentation and welcomes your comments and suggestions. To share your comments, go to <http://docs.sun.com> and click Send Comments. In the online form, provide the full document title and part number. The part number is a 7-digit or 9-digit number that can be found on the book's title page or in the document's URL. For example, the part number of this book is 819-4726.

Overview

This chapter provides a description of the tools, methods, and information sources available for troubleshooting the Sun Java System Application Server Server 8.2 Platform Edition. Guidelines for evaluating and investigating a problem are included.

- “Planning Ahead” on page 11
- “Identifying the Problem” on page 12
- “Seeking a Solution” on page 14

Planning Ahead

As applications get deployed, undeployed, and redeployed, and as you experiment with different server configuration settings, there may be times when your server gets into a confused or unstable state. In such cases, it is useful to have a previously saved working configuration on which to fall back. This is not problem solving, per se, but rather a way to avoid problems in the first place.

The Application Server `asadmin` command includes a `backup-domain` option that backs up the domain(s) you specify. Use this option to take periodic “snapshots” of your server configuration. Then, if necessary, use the `restore-domain` option to restore one or more domains to a known working state.

Refer to the *Application Server Administration Guide* for complete instructions on using the `asadmin backup-domain` and `restore-domain` options. Briefly, however, for the purposes of this *Troubleshooting Guide*, use the following procedure to backup and restore a server configuration:

▼ To back up and restore a server configuration

Steps 1. Start the Application Server.

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

2. Stop the domain.

```
install_dir/bin/asadmin stop-domain domain_name
```

3. Back up the domain.

```
install_dir/bin/asadmin backup-domain domain_name
```

Backed up directories are stored by default in the *install_dir/backups* directory.

4. Make changes to the Application Server configuration and/or domain(s), as desired.

5. If necessary, restore the server and/or domain configuration to the state saved in Step 3, above.

```
install_dir/bin/asadmin restore-domain --filename backup_file domain_name
```

Identifying the Problem

J2EE application servers are typically deployed in complex and highly sophisticated operating environments. The Sun Java System Application Server covers a broad range of technologies, including Java, Java servlets, XML, JSP, JDBC data sources, EJB technology, and more. Other products and tools associated with the Application Server are LDAP, Web Server, SunONE Message Queue, deployment and migration tools, and so on. Understanding and diagnosing complex issues involving so many disparate components requires thorough knowledge and a careful diagnostic process.

Gathering any or all of the following information will make it easier to classify a problem and search for solutions. Note that operating system utilities, such as `pkginfo` and `showrev` on Solaris and `rpm` on Linux, are helpful in gathering system information.

▼ Verifying configuration information

Steps 1. What are the exact version numbers of the operating system and products installed?

2. **Have any patches been applied? If so, specify product and operating system patch numbers.**
3. **How is the system configured?**
4. **What system resources does the system have (memory, disk, swap space, and so on)?**
5. **How many application servers, web servers, and directory servers are installed?**
6. **How is the web server connected to Application Server? On the same machine or not?**
7. **How is the Application Server connected to the directory server?**
8. **Are application servers in a cluster or not?**
9. **Was any upgrade done? If so, what were source and target versions?**
10. **Was a migration done? If so, what were source and target versions?**
11. **Have any new applications been deployed?**
12. **Is SSL enabled or not?**
13. **What versions of the HADB and the backend database are being used?**
14. **What JDBC driver is being used to access the database?**
15. **What JDK version is being used?**
16. **What are the JVM heap, stack, and garbage collection-related parameters set to?**
17. **What are the JVM options?**
18. **What third-party technologies are being used in the installation?**
19. **Are the interoperating component versions in compliance with the compatibility matrix specified in the release notes?**

After gathering this information:

- Collect web server error and access log data (web server instance-specific).
- Collect any Application Server stack traces. Note that a fresh set of logs associated with the specific problem should be run. This avoids scanning gigabytes of irrelevant log information.
- Determine the sequence of events that occurred when the problem first appeared, including any steps that may already have been taken to resolve the problem.

Seeking a Solution

After identifying the problem and formulating a preliminary hypothesis of what may be wrong, you are ready to do some investigation.

The following topics are addressed in this section:

- “Verify System Configuration” on page 14
- “Evaluate Messages” on page 14
- “Examine Log Files” on page 15
- “See if the Problem has been Solved Before” on page 18
- “Search the Product Documentation” on page 18
- “Search the Knowledge Base” on page 18
- “Search or Participate in the Online Forum” on page 19
- “Contact Support” on page 19

Verify System Configuration

Sometimes the most obvious solutions are overlooked, and so the first step is to verify the system configuration. Refer to the *Sun Java System Application Server 8.2 Release Notes* for the most up-to-date system requirements and dependencies.

Evaluate Messages

Messages generally include information about the attempted action, the outcome of the action, and, if applicable, the cause of jeopardy or failure.

Types of Messages

The log files contain the following general types of message entries:

- **Error** – These messages mark critical failures that cause status to be reported as Failed. Error messages generally provide detailed information about the nature and the cause of the problem that occurred.
- **Warning** – These messages mark non-critical failures. Warning messages generally contain information about the cause and the nature of the failure, and also provide possible remedies.
- **Information** – These messages mark normal completion of particular tasks.

Error Messages

A problem is often accompanied by an error message that prevents the application from proceeding.

- In some cases, the message is very clear about what is wrong and what needs to be done, if anything, to fix it. For example, if you start a domain using the `asadmin start-domain` command, then inadvertently issue the same command again after the domain has started, the following message is displayed:

```
userD:\\Sun\\studio5_se\\appserver8\\bin\\>asadmin start-domain
Domain already started : domain1
Domain domain1 Started.
```

In this case, the message gives clear guidance and the problem can be disregarded.

- Sometimes an error message gives only general information about the problem or solution, or suggests multiple possibilities. For example:

```
[16/Jun/2003:22:20:50] SEVERE ( 2204): WEB0200: Configuration error
in web module [JAXBProjectStudio] (while initializing virtual server
[server1]) com.ipplanet.ias.config.ConfigException: Failed to load
deployment descriptor for: JAXBProjectStudio cause:
java.io.FileNotFoundException:
```

In this case, the problem is not obvious, or there might be multiple things wrong. You might have to consider various possibilities and perhaps a number of solutions. If the proposed fix is time consuming or costly, take steps to ensure that the fix is likely to be correct before actually doing anything.

- Some error messages are either not helpful or provide little guidance; for example:

```
[23/Jun/2003:16:50:45] WARNING ( 1972): for host 127.0.0.1 trying to GET
/SupplierServiceClient1/SupplierServiceClient1_SOAP.html, send-file
reports: HTTP4144: error sending D:/Sun/studio5_se/appserver8/domains/
domain1/server1/applications/j2ee-modules/SupplierServiceClient1_1/
SupplierServiceClient1_SOAP.html (Overlapped I/O operation is in progress.)
status=1:5
```

In this case, there is very little information to go on. It is especially important to identify the exact situation that caused the error, and what the symptoms are before proceeding.

For descriptions of all the Application Server error messages, refer to the *Sun Java System Application Server Error Message Reference* at the following location:

http://docs.sun.com/db/coll/ApplicationServer8_04q4

Examine Log Files

A number of the Application Server subsystems create log files and log their events to these files. The primary purpose of these log files is to provide troubleshooting information.

In addition to the message text, a logged message provides the following information:

- Date and time of the event

- Log level for the event — Application Server-specified log level ID or name
- Process identifier (PID) — PID of the Application Server process
- (optional) Virtual server identifier (VSID) — VSID that generated the message
- Message identifier (MID) — subsystem and a four digit integer
- Message data

The specific logs associated with each Application Server problem area are discussed in the associated chapters of this manual.

Log Levels

The Application Server has many log levels that can be set in the Administration GUI (FINEST, FINER, FINE, CONFIG, INFO, WARNING, SEVERE, ALERT, and FATAL). All messages are logged when the log level is set to FINEST and only serious error messages appear if the log level is set to FATAL.

Note that the more detailed log levels (FINEST, FINER, FINE) can generate high volumes of log information for certain events, which may make it appear at first glance that there is an error condition when in fact there is not.

All messages with a log level less than the default level of INFO (FINEST, FINER, FINE, and CONFIG) provide information related to debugging and must be specifically enabled. Instructions for doing this are contained in the *Sun Java System Application Server Administrator's Guide*.

In addition to the standard JDK log levels, the Application Server has added log levels designed to map more intuitively to the Application Server log file (`server.log`) and to tightly integrate with Solaris. The log levels ALERT and FATAL are specific to the Application Server and are not implemented in the JDK1.4 logging API.

Note – For information on the event log mechanism used in the Microsoft Windows operating environment, refer to the Windows help system index using the keywords Event Logging. If you choose to send logs to the Windows `server.log` file, only messages with a log level of INFO, WARNING, SEVERE, ALERT, or FATAL are logged to the Windows Event Log.

Log Options

The Administration GUI provides the following two logging options:

- **Option 1** — Log `stdout` (`System.out.print`) content to the event log
- **Option 2** — Log `stderr` (`System.err.print`) content to the event log

When these options are set, `stdout` and `stderr` messages are written to the `server.log` file. (The event log is a `syslog` daemon on Solaris and Event Log on Microsoft Windows.)

If the above options are not set:

- Anything written to `stdout` or `stderr` (that is, using `System.out` or `System.err`) will not appear in the logs.
- Messages logged with the JDK logger will appear in the logs.
- Messages written to `stdout` or `stderr` appear with the `INFO` level, but do not have a message ID.

Client Side Logging

The Application Client Container (ACC) has its own log service and can only log to a local file. The ACC typically runs in its own process, on a different host from the Application Server. It has its own logging infrastructure and its own log file. The `sun-acc.xml` file contains the ACC configuration. Refer to the *Sun Java System Application Server Developer's Guide* for more information.

Obtaining a Thread Dump

This section explains how to obtain a thread dump for Application Server 8.2. By default, the server dumps a core file and restarts with the `-Xrs java-option` flag in the `server.xml` file.

On UNIX

The following procedure describes how to obtain a server thread dump on UNIX.

▼ To obtain a server thread dump on UNIX

- Steps**
1. Verify that the `server.xml` file for the affected server instance does not include the `-Xrs java-option` flag. Remove the `-Xrs java-option` flag if it exists.
 2. If the option is changed, restart the server instance.
 3. Use the `ps` command to determine the `java` and/or `appservDAS` processes under which the application server is running.
 4. Run the following command on the application server instance:

```
kill -3 pid
```

The `kill` command redirects the thread dump to the `server.log` file for server the instance.

On Windows

The following procedure describes how to obtain a server thread dump on Windows.

▼ To obtain a server thread dump on Windows

- Steps**
1. Verify that the `server.xml` file for your server instance does not include the `-Xrs java-option` flag. Remove the `-Xrs java-option` flag if it exists.
 2. If the option was changed, restart your Application Server.
 3. Type `ctrl-brk` in the Application Server window. The thread dump will be redirected to the `server.log` file for the instance.

See if the Problem has been Solved Before

A good initial step is to scan this Troubleshooting Guide to see if the problem is addressed here. If so, select the appropriate solution. Many of the solutions contain references to other documents in the Application Server document collection for additional details, explanations, or examples.

Search the Product Documentation

Start by reading the Release Notes for the version of the product you are troubleshooting.

The documentation for this Application Server product release is available at:

<http://docs.sun.com/db/coll/ApplicationServer81>

Descriptions of the Application Server manuals are listed in “[Application Server Documentation Set](#)” on page 4.

Search the Knowledge Base

The Knowledge Base is a collection of articles on product issues that provide information helpful for troubleshooting. To access the Knowledge Base:

▼ To search the Knowledge Base

- Steps**
1. Go to SunSolve (<http://sunsolve.sun.com/pub-cgi/show.pl?target=home>).
 2. Under SunSolve Collections, click the Search Collections link.

3. **Select the checkbox for the collection(s) to search.**
4. **Click Next.**
5. **Enter the search criteria.**
6. **Click Go.**

Search or Participate in the Online Forum

Browse directly in any of the online forums, or log in and register to start posting messages. The Application Server online forum is available at:
<http://swforum.sun.com/jive/index.jsp?cat=7>

Contact Support

When necessary, gather together the information you have acquired and contact technical support at <http://www.sun.com/service/contacting>.

Common Problems

This section covers the most common problems you may encounter when using the Application Server:

- “`asadmin start-domain` Command Fails” on page 21
- “Automatically Restarting a Domain or Node Agent on Reboot” on page 23
- “Cannot Find Log Files” on page 26
- “Accessing Local Server Fails (`http://localhost:8080`)” on page 28
- “Accessing Remote Server Fails” on page 29
- “Cannot Access the Administration Console” on page 31
- “Cannot Access a Server Application” on page 33
- “Server Will Not Start on Windows (Port Conflict)” on page 35
- “Port Conflicts Debugging Multiple Instances on the Same Server” on page 36
- “Two Server Instances Bind to Same Port on Windows” on page 37
- “Error: System cannot find the path specified” on page 38
- “Application Generates Error
`persistence.support.JDODataStoreException`” on page 39
- “Using `asadmin set` Command May Produce Unexpected Results” on page 39
- “Cannot Undeploy Or Redeploy Application With Open Streams to Jar Files (Windows Only)” on page 40
- “Cannot Reinstall Application Server After Manual Deletion of Directories” on page 41
- “Cannot Produce a JVM Thread Dump After Server Crash” on page 41

`asadmin start-domain` Command Fails

The command `asadmin start-domain` fails with one of the following errors:

- “Error: CLI143 There is more than one domain...” on page 22

- [“Error: Could Not Start Domain” on page 22](#)

Error: CLI143 There is more than one domain...

Description

When issued with no arguments, the command `asadmin start-domain` fails with the error:

```
CLI143 There is more than one domain in C:\\Sun\\AppServer\\domains.  
Please use operand to specify the domain.  
CLI156 Could not start the domain null.
```

This error occurs when there is more than one domain in the `domains` directory, none of them is named `domain1`, and no domain is specified with the `start-domain` command.

Solution

Specify the domain when issuing the `start-domain` command:

```
asadmin start-domain domain1
```

Error: Could Not Start Domain

Description

This message comes from Application Server 8. The full message looks like either:

```
Could not start the domain.  
There are no domains.
```

or

```
Could not start the domain.  
No default domain. Need to enter a domain.
```

This error occurs when Application Server 8 is installed on the same system, and its `asadmin` command (at `/usr/sbin`) is found on the `PATH` before the `asadmin` command for Application Server 8 at `install_dir/bin`. The situation is especially likely to occur on Solaris/Linux systems when `.` is not specified as part of the `PATH` variable. Without `.` in the `PATH`, the `asadmin` command in `/usr/sbin` is found first, even when the current directory is `install_dir/bin`.

Solution

Make sure *install_dir/bin* is in the PATH ahead of */usr/sbin*, or that *.* is in the PATH ahead of */usr/sbin* if you access *asadmin* by changing directories to *install_dir/bin*. Alternatively, if you do change to *install_dir/bin* to run *asadmin*, be sure to include *./* in the command name; for example:

```
cd install_dir/bin
./asadmin
```

Automatically Restarting a Domain or Node Agent on Reboot

If your domain or node agent is stopped unexpectedly (for example, if you need to restart your machine), you can configure your system to automatically restart the domain or node agent on reboot.

- [“Restarting Automatically on UNIX Platforms” on page 23](#)
- [“Restarting Automatically on the Microsoft Windows Platform” on page 24](#)

Restarting Automatically on UNIX Platforms

Restarting a Domain

To restart your domain on a UNIX platform, add a line containing the appropriate *asadmin start-domain* command to your */etc/inittab* file. If you use */etc/rc.local*, or your system’s equivalent, place a the desired *asadmin* command in */etc/rc.local*.

For example, to restart *domain1* for an Troubleshooting Guide installed in the */opt/SUNWappserver* directory, using a password file called *password.txt*, add the following line to */etc/inittab* or */etc/rc.local*:

```
das:3:respawn:/opt/SUNWappserver/bin/asadmin start-domain --user admin
--passwordfile /opt/SUNWappserver/password.txt domain1
```

Be sure to put the text on one line. The first three letters are a unique designator for the process and can be altered.

Restarting a Node Agent

To restart a node agent, the syntax is similar. For example, to restart `agent1` for an Troubleshooting Guide installed in the `/opt/SUNWappserver` directory using a password file called `password.txt`, add the following line to `/etc/inittab` or `/etc/rc.local`:

```
das:3:respawn:/opt/SUNWappserver/bin/asadmin start-node-agent --user admin
--passwordfile /opt/SUNWappserver/password.txt agent1
```

Restarting Automatically on the Microsoft Windows Platform

To restart automatically on Microsoft Windows, create a Windows Service. Use the `appservService.exe` and `appserverAgentService.exe` executables shipped with Sun Java System Application Server in conjunction with the Service Control command (`sc.exe`) provided by Microsoft.

- The `sc.exe` command comes with Windows XP and is either located in the `C:\windows\system32` directory or `C:\winnt\system32` directory.
- As of this writing, the Windows 2000 `sc.exe` is available for download at: <ftp://ftp.microsoft.com/reskit/win2000/sc.zip>. For more information on using `sc.exe`, see: http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dndllpro/html/msdn_scmslite.asp.

Use `appservService.exe` and `appservAgentService.exe` as follows:

```
C:\winnt\system32\sc.exe create service_name binPath= \\\"fully_qualified_
path_to_appservService.exe \\\"fully_qualified_path_to_asadmin.bat
start_command\\\"
\\\"fully_qualified_path_to_asadmin.bat stop_command\\\" start= auto
DisplayName= \"display_name\"
```

Starting and Stopping a Domain

To create a service called `SunJavaSystemAppServer DOMAIN1` that starts and stops `domain1` using password file `C:\Sun\AppServer\password.txt`, run the following command:

```
C:\windows\system32\sc.exe create domain1 binPath= \"C:\Sun\AppServer\
lib\appservService.exe \\\"C:\Sun\AppServer\bin\asadmin.bat start-domain
--user admin --passwordfile C:\Sun\AppServer\password.txt domain1\\\"
\\\"C:\Sun\AppServer\bin\asadmin.bat stop-domain domain1\\\"
start=auto DisplayName= \"SunJavaSystemAppServer DOMAIN1\"
```

Starting and Stopping a Node Agent

To create a service that starts and stops the node agent `agent1`, run the following command:

```
C:\windows\system32\sc.exe create agent1 binPath= "C:\Sun\AppServer\lib\appservAgentService.exe" \\ "C:\Sun\AppServer\bin\asadmin.bat"
start-node-agent --user admin --passwordfile C:\Sun\AppServer\password.txt agent1 \\ "C:\Sun\AppServer\bin\asadmin.bat"
stop-node-agent agent1 \\ "" start=auto DisplayName="SJESAS_SE8.1 AGENT1"
```

Note – The start and stop commands entered as part of the `binPath=` parameter must have the correct syntax. To test, run the commands from the command prompt. If the commands do not properly start or stop the domain or node agent, the service will not work correctly.

Also, do not use a mixture of `asadmin start` and `stop` commands and `service start` and `stops`. Mixing the two can cause the server status to be out of sync. For example, the service might not show that the component has started even though the component is not running. To avoid this situation, always use the `sc .exe` command to start and stop the component when using services.

Security for Automatic Restarts

Handle the password and master password required when starting in one of the following ways:

- On Microsoft Windows, configure the service to ask the user for the password.
 - In the Services Control Panel, double-click the service you created.
 - In the Properties window, click the Log On tab.
 - Check “Allow service to interact with desktop” to prompt for the required passwords when starting the component.

You have to log in to see the prompts, and entries are not echoed back as you type them. This method is the most secure way to use the services option, but user interaction is required before the service becomes available.

If the “interact with desktop” option is not set, the service stays in a “start-pending” state and appears to hang. Kill the service process to recover from this state.
- On Windows or UNIX, create a domain using the `--savemasterpassword=true` option and create a password file to store the admin password. When starting the component, use the `--passwordfile` option to point to the file that contains the password. The admin password can also be added by using the `--password` option with the `asadmin start` command. Be aware that this method is less secure because the admin password is stored in clear text.

For example:

- Create a domain with a saved master password. Using this syntax, you are prompted for the admin password and master password:

```
asadmin create-domain --adminport 4848 --adminuser admin
--savemasterpassword=true --instanceport 8080 domain1
```

- On Windows, create a service using a password file to populate the admin password:

```
C:\windows\system32\sc.exe create domain1 binPath=
"C:\Sun\AppServer\lib\appservService.exe
\\"C:\Sun\AppServer\bin\asadmin.bat start-domain --user admin
--passwordfile C:\Sun\AppServer\password.txt domain1\\"
\\"C:\Sun\AppServer\bin\asadmin.bat stop-domain domain1\\"
start= auto DisplayName= "SJESAS_PE8.1 DOMAIN1"
```

The path to the password file `password.txt` is `C:\Sun\AppServer\password.txt`. It contains the password in the following format:

```
AS_ADMIN_password=password
```

For example, for a password `adminadmin`:

```
AS_ADMIN_password=adminadmin
```

- On UNIX, use the `--passwordfile` option in the line you add to the `inittab` file:

```
das:3:respawn:/opt/SUNWappserver/bin/asadmin start-domain
--user admin --passwordfile /opt/SUNWappserver/password.txt domain1
```

The path to the password file `password.txt` is `/opt/SUNWappserver/password.txt`. It contains the password in the following format:

```
AS_ADMIN_password=password
```

For example, for a password `adminadmin`:

```
AS_ADMIN_password=adminadmin
```

- Creating a service using a password that is populated from a command line option:

```
C:\windows\system32\sc.exe create domain1 binPath= "C:\Sun\AppServer\
lib\appservService.exe \\"C:\Sun\AppServer\bin\asadmin.bat start-domain
--user admin --password adminadmin domain1\\" \\"C:\Sun\AppServer\bin\
asadmin.bat stop-domain domain1\\" start=auto
DisplayName="SJESAS_PE8.1 DOMAIN1"
```

Cannot Find Log Files

The following Application Server logs are useful for troubleshooting installation problems:

- **Server log file** — For troubleshooting server configuration and deployment problems

- **HTTP server access logs** — For troubleshooting HTTP server problems, and for tracing the activity of HTTP requests entering the Application Server instances

Both the installation and uninstallation programs create log files and log all installation and uninstallation events to these files. The primary purpose of these log files is to provide troubleshooting information.

In addition to installation program messages and log files, operating system utilities such as `pkginfo` and `showrev` on Solaris and `rpm` on Linux can be used to gather system information.

Log file entries include information about the attempted action, the outcome of the action, and, if applicable, the cause of failure. The log files contain the following types of message entries:

- **INFO** — These messages mark normal completion of a particular installation tasks.
- **WARNING** — These messages mark non-critical failures. Warning messages generally contain information about the cause and the nature of the failure, and also provide possible remedies.
- **ERROR** — These messages mark critical failures that cause installation or uninstallation status to be reported as Failed. Error messages generally provide detailed information about the nature and the cause of the problem that occurred.

The domain-specific logs are located in `install_dir/domains/domain1/logs/`. Log files for the server installation in general are located as follows:

- **Solaris, root user installation/uninstallation:**

`/var/sadm/install/logs`

- **Solaris, non-root installation/uninstallation:**

`/var/tmp`

- **Linux installation/uninstallation:**

`/var/tmp`

- **Windows installation/uninstallation:**

`%TEMP%`

Log file names are different for each product distribution, but are platform independent:

- Platform Edition, both SDK and Application Server distributions, main installation/uninstallation log files:

`Install_Application_Server_8PE_<timestamp>.log`
`Uninstall_Application_Server_8PE_<timestamp>.log`

- Platform Edition, Application Server only distribution, low-level installation/uninstallation log files:

`Sun_Java_System_Application_Server_Platform_Edition_install.B<timestamp>`
`Sun_Java_System_Application_Server_Platform_Edition_uninstall.B<timestamp>`

- Platform Edition, SDK distribution, low-level installation/uninstallation log files:

`Java_2_Platform_Enterprise_Edition_1.4_SDK_install.B<timestamp>`
`Java_2_Platform_Enterprise_Edition_1.4_SDK_uninstall.B<timestamp>`

- Enterprise Edition, main installation/uninstallation log files:

```
Install_Application_Server_8EE_<timestamp\>.log  
Uninstall_Application_Server_8EE_<timestamp\>.log
```

- Enterprise Edition, low-level installation/uninstallation log files:

```
Sun_Java_System_Application_Server_Enterprise_Edition_  
install.B<timestamp\>  
Sun_Java_System_Application_Server_Enterprise_Edition_  
uninstall.B<timestamp\>
```

Accessing Local Server Fails (<http://localhost:8080>)

Things to check for this error include the following:

- [“Did the Server Start?” on page 28](#)
- [“Was the Server Started at the Expected Port?” on page 29](#)

Did the Server Start?

Description

If the console window is still open, the expected message is:

```
Domain domain Started
```

where *domain* is the name of the default domain. This indicates that the default domain was started successfully.

If the console window is already closed, check for messages in the log file:

```
install_dir/domains/domain1/logs/server.log
```

If startup was successful, the expected message is similar to that on the console, and appears at the end of the log file:

```
[INFO] [...] [...] [date&time] [Application server startup complete .]
```

Was the Server Started at the Expected Port?

Description

The server might be running at a different port number than expected, either because it was intentionally installed there, or because another server was already running on the default port when the server was installed.

▼ To determine which port number the server is actually using

Steps 1. **Examine the server's configuration file:**

```
install_dir/domains/domain1/config/domain.xml
```

2. **Find the `http-listener` element.**

3. **Inspect the value of the `port` attribute.**

Be sure to enter the correct port number when invoking the server.

Note – The server's default port number is 8080, however, there are a number of ways in which the expected value can change:

- A different port number was specified during installation.
 - A previous installation exists.
 - If the specified port number is already taken by another application when the server is started, the port number rolls forward to the next available number. For example, if a server is already running on the default 8080 port, the new Application Server instance uses port number 8081. If two servers are running, the port number rolls to 8082, and so on.
-

Accessing Remote Server Fails

When attempting to open the start page of the Application Server, the initial screen does not appear.

Things to check include the following:

- [“Is the Server Available Locally?”](#) on page 30

- [“Is the Proxy Setting Causing a Problem?”](#) on page 30

Is the Server Available Locally?

Description

If the server cannot be accessed from the web, but it is running locally, then the server is actually running.

Solution

Verify that the server is running locally.

▼ To verify that the server is running locally

Steps 1. **Log on to the machine where the server is running.**

2. **Go to the local web page. For example, if 8080 is the default port, go to:**

```
http://localhost:8080/
```

If the start page does appear, there is a problem with the web connection that prevents accessing the server remotely. If the start page does not appear, see [“Did the Server Start?”](#) on page 28

Is the Proxy Setting Causing a Problem?

Description

The server should be accessible directly from the host on which it is running (localhost); for example, using the default port, 8080:

```
http://localhost:8080/
```

Solution

A server instance running on localhost may not be accessible if the server host machine is connected to the web through a proxy. To solve this problem, do one of the following:

- Set the browser to bypass the proxy server when accessing localhost. Refer to the browser's help system for information on how to do this.
- Use the fully-qualified host name or IP address of your system; for example:
`http://myhost.mydomain.com:8080/`

Note – To find the host name and domain for the localhost machine:

- **On Microsoft Windows** — On the desktop, right-click My Computer and select Properties from the pop-up menu. A System Properties dialog is displayed. Click Network Identification to see the computer name.
 - **On Solaris or Linux** — Type `hostname` at the command prompt.
-

Cannot Access the Administration Console

The Administration Console provides an interface for administrative functions. If the Administration Console is not accessible, it might be for one of several reasons.

- [“Is the Application Server Running?” on page 31](#)
- [“Is the Administration Console Running on the Expected Port?” on page 32](#)
- [“Is the Security Manager Disabled?” on page 32](#)

Is the Application Server Running?

Description

The server must be running before the Administration Console can be accessed.

Solution

Review the information in [“Did the Server Start?” on page 28](#) to determine if the server is running.

Is the Administration Console Running on the Expected Port?

Description

The default port number for the EE and SE Administration Console is 4849; for the PE Administration Console it is 4848. Also note that the URL for the EE and SE console requires secure HTTP (`https://servername:4849/`), whereas the PE console uses standard HTTP (`http://servername:4848/`). However, it could be running on a different port number than expected, either because it was intentionally installed there, or because that port was taken when the server was started.

Solution

Refer to [“Was the Server Started at the Expected Port?”](#) on page 29 for guidelines on verifying the port on which the Administration Console is running, and be sure to enter the correct port number and HTTP protocol when invoking the Administration Console.

Is the Security Manager Disabled?

Description

According to the *J2EE 1.4 Specification*, the Security Manager is not optional; it must be enabled in the Application Server. Since there is no configuration interface in the Application Server for disabling the Security Manager, it can only be disabled by directly modifying the `domain.xml` configuration file in such a way that the following line is removed:

```
<jvm-option\>-Djava.security.policy=yourPolicy</jvm-option\>
```

Solution

The `-Djava.security.policy=yourPolicy` option must be present in the `domain.xml` file to access the Administration Console.

Cannot Access a Server Application

If a particular application cannot be accessed through the Application Server, some things to check include the following:

- “Is the Application Server Running?” on page 33
- “Was Application Deployment Successful?” on page 33
- “Invalid User or Password When Using Don’ t Prompt Option” on page 33
- “Administrator User Name or Password Not Known” on page 34

Is the Application Server Running?

Description

If the Application Server is not running, applications will not be accessible.

Solution

Review the information in “Did the Server Start?” on page 28 to determine if the server is running. The server must be running before a server application can be accessed.

Was Application Deployment Successful?

Description

An application must be successfully deployed before it can be accessed.

Solution

Check the server’s log file:

```
install_dir/domains/domain1/server.log
```

Invalid User or Password When Using Don’ t Prompt Option

You are getting the error, Invalid user or password, but you installed the system with the Don’ t Prompt option, so the password should be supplied automatically.

Description

The correct password may not have been specified during installation, or it may not be passed when the domain is started.

Solution

Check the password in the `.asadminprefs` file. On UNIX/Linux systems, it is in the home directory for the user under which the server was installed. On Windows, it is in `C:\Documents and Settings\username`. The contents look something like this:

```
AS_ADMIN_USER=admin
AS_ADMIN_PASSWORD=administrator
```

Administrator User Name or Password Not Known

If you have forgotten the administrator user name, you can find it by inspecting the `.adminprefs` file, as described in the section above, or by inspecting the `install_dir/domains/domain1/config/keyfile`, where `domain1` is the default domain. For a different domain, substitute its name in the path.

If you have forgotten the administrator password, you will need to create a new user name-password pair by removing the user name and password, creating new ones, and restarting the server. (You will not be able to read the password, because it is encrypted in the keyfile.)

▼ To remove the user name and password completely

Steps 1. Stop the server, if it is currently running.

2. Change to the appropriate **WEB-INF** directory; for example:

```
install_dir/lib/install/applications/adminapp/adminapp_war/WEB-INF
```

3. Comment out the entire `<security-constraint>` element in the `web.xml` file.

Do not delete the element, as you will be reenabling it later. This action disables security for command-line operations.

Note – The commands will still expect a value for `--username` (or `-u`) and `--password` (or `-w`). But these can be dummy values, since the server side does not impose any security.

4. Start the server.

At this point, the server does not have command-line security.

5. Run the following command:

```
asadmin create-file-user --user <dummy\> --password <dummy\>
--userpassword <new_secret\> --groups asadmin <new_user_id\>
```

This command creates the following new entry:

```
<install_dir\>/domains/domain1/config/keyfile
```

6. Uncomment the `<security-constraint\>` element in `web.xml` file.

7. Restart the server for the new user name-password to take effect.

Note – When the server is started, any remote command-line operations will need `new_user_id` and `new_secret` as user name and password.

Server Will Not Start on Windows (Port Conflict)

If a message similar to the following is displayed when starting the Application Server on Microsoft Windows, a server port conflict has occurred:

```
Address already in use
```

This error occurs when another application is running on the Application Server port (default 8080), or because a previous instance of the Application Server did not shut down cleanly.

Other things to check include the following:

- [“Is Another Application Running on the Server’s Port?” on page 36](#)
- [“Has an Ungraceful Shutdown Occurred on a Previously Running Server?” on page 36](#)

Is Another Application Running on the Server's Port?

If another application is using the server's port, stop the other application, then restart the Application Server.

Note – The installer attempts to avoid port conflicts by choosing the next available port when the default port is in use—but that only works if application using the default port was running when the Application Server was installed.

Has an Ungraceful Shutdown Occurred on a Previously Running Server?

Use the `asadmin stop-domain` command to stop the server, or explicitly kill the Java process and then restart the Application Server.

Port Conflicts Debugging Multiple Instances on the Same Server

Description

Port conflict errors can occur when debugging multiple instances on the same server that are part of the same cluster.

Solution

Modify the `domain.xml` file to remove the `address` attribute from the `-Xrunjdpw` option in the `java-config` element for the cluster. This results in the JVM choosing a random debug port for the instance. The port number chosen for the instance is displayed in the server log when it is started; for example:

- **Before:**

```
debug-options="-Xdebug -Xrunjdpw:transport=dt_socket,server=y,
suspend=n,address=9009"
```

- **After:**

```
debug-options="-Xdebug -Xrunjdp:transport=dt_socket,server=y,  
suspend=n"
```

Two Server Instances Bind to Same Port on Windows

Description

This problem only occurs on Windows 2000/XP systems with the Application Server Enterprise Edition (not Platform Edition) software, and is due to a known Windows security flaw rather than a problem with the Application Server itself.

The problem occurs when two or more instances of the Application Server are created using the same port number for the `instanceport` option; for example:

```
asadmin create-domain -adminport 5001 <options> -instanceport 6001  
<domain\  
asadmin create-domain -adminport 5002 <options> -instanceport 6001  
<domain\  
>
```

When the two domains are started on a UNIX/Linux system, a port conflict error is thrown and the second instance fails to start. However, when the two domains are started on Windows 2000/XP, no error is thrown, both server instances start, but only the first instance is accessible at the specified port. When that first server instance is subsequently shut down, the second instance then becomes accessible. Moreover, when both instances are running, the Windows `netstat` command shows the duplicate listeners as active, but only the first listener can respond to requests.

Solution

Be sure to use unique port numbers for all server instances on Windows systems.

Error: System cannot find the path specified

Description

This error message occurs when attempting to start the server after deleting the J2SE directory that was specified during installation. This situation generally occurs after being informed during the install that the J2SE platform needs an upgrade, and the upgrade takes place after the Application Server installation.

Solution 1

To use the new J2SE for all domains, change the `AS_JAVA` variable in `asenv.conf` (Solaris/Linux), or `asenv.bat` (Windows).

Solution 2

The J2SE version can be changed on a per-domain basis by modifying the `java-home` attribute for the `java-config` element in the domain's `domain.xml` file.

```
<java-config ...  
java-home="path"  
... \>
```

Solution 3

A more time-intensive solution is to uninstall and then reinstall the server.

Application Generates Error `persistence.support. JDODataStoreException`

Description

A `com.sun.jdo.api.persistence.support.JDODataStoreException` is generated by an application, with a nested `java.sql.SQLException` indicating a duplicate primary key.

Even if the application is checking for a `CreateException`, it does not see one. The Enterprise JavaBeans specification requires a `CreateException` to be thrown only if two beans with the same primary key are created in the same transaction, so a `CreateException` is not thrown on transaction rollback if two entity beans with duplicate primary keys are created.

Solution

If an application creates an entity bean with a duplicate primary key, check to see if the primary key exists by calling `findByPrimaryKey` before calling `create`.

Using `asadmin set` Command May Produce Unexpected Results

Description

Unexpected results are returned when setting variables in a command, such as:

```
asadmin set name=${a-b}
```

In this case, `name` is set to `b`, not `{a-b}` because the shell syntax `${a=b}` is interpreted as “if the variable `a` is unset, substitute the value `b`, otherwise substitute the value of `a`.” This is standard shell behavior. For example, consider the following:

```
asadmin set default-config.http-service.http-listener.http-listener-1.port=
${http-listener-1-port}
```

In this case,
default-config.http-service.http-listener.http-listener-1.port is
set to listener-1-port, which is invalid.

Cannot Undeploy Or Redeploy Application With Open Streams to Jar Files (Windows Only)

Description

On Windows systems, after running an application, subsequent attempts to undeploy it or redeploy it throw exceptions about the server being unable to delete a file or rename a directory.

On Windows systems, an application may use `getClass().getResource` or `getResourceAsStream` methods to locate a resource inside the application, particularly in jar files that are in the application or accessible to it. If the streams remain open, subsequent attempts to redeploy or undeploy the application can fail. In addition, the Java runtime by default caches streams to jar files for performance reasons.

Solution

Be sure to close streams opened by your applications. Also, if an application needs to be redeployed or undeployed repeatedly, and also needs to obtain a resource from a jar file using `getResource` or `getResourceAsStream`, consider using `getClass().getResource` which returns a URL object, then invoke the `url.setUseCaches` method to turn off caching for that jar file, and then use `url.getInputStream()` to obtain the stream.

Although turning off caching for accesses to the jar file can slow down performance, this approach does allow the application to be undeployed or redeployed. Note also that if the `getClass().getResourceAsStream` method is used instead, then the jar file in which the resource is located will be cached (this is the default Java runtime setting) and remain open until the server is stopped.

Cannot Reinstall Application Server After Manual Deletion of Directories

Description

If the Application Server directories are deleted manually rather than by means of the included `uninstall` the program, subsequent attempts to reinstall the Application Server in the same directory fail. This is because the installation directory information stored in `/tmp/productregistry` file remains even though the program directories have been removed.

Solution 1

Remove Application Server directory information from the `<location\>` property entries in the `/tmp/productregistry` file; for example, change:

```
<location\>/opt/SUNWappserver/jdk</location\>
```

to

```
<location\></location\>
```

Solution 2

Reinstall the Application Server in a different directory.

Cannot Produce a JVM Thread Dump After Server Crash

Description

If the Application Server crashes, the server dumps a core file and, by default, restarts with the `-xrs` flag, which prevents the dump of a JVM thread dump.

Solution

▼ To enable a JVM thread dump

- Steps**
1. Comment out the `-Xrs` flag in the `server.xml` file for the Application Server.
 2. Kill the server process (`kill -3` on UNIX; Ctrl+Break on Windows).

Security Problems

This chapter covers problems that you may encounter as a result of security settings:

- “`java.security.AccessControlException: Access Denied Error`” on page 43
- “`javax.ejb.AccessLocalException: Client Not Authorized Error`” on page 44
- “Authentication is Not Working With the Solaris Realm” on page 44
- “Mutual Authentication Not Working With the Application Client” on page 45

`java.security.AccessControlException: Access Denied Error`

Description

The following error occurs from an application client or in the `server.log`:

```
java.security.AccessControlException: access denied
(java.util.PropertyPermission name write...
```

There is a permissions issue in the policy files. Either the `client.policy` for the application client or the `server.policy` for server side components does not have permission to set the property.

Solution

Add the permission in `client.policy` (for the application client), or in `server.policy` (for EJB/web modules) for the application that needs to set the property. By default, applications only have “read” permission for properties.

For example, to grant read/write permission for all the files in the codebase directory, add or append the following to `client.policy` or `server.policy`:

```
grant codeBase "file:/.../build/sparc_SunOS/sec/-" {
    permission java.util.PropertyPermission "*", "read,write";
};
```

javax.ejb.AccessLocalException: Client Not Authorized Error

Description

Role-mapping information is available in Sun-specific XML (for example, `sun-ejb-jar.xml`), and authentication is okay, but the following error message is displayed:

```
[...INFO|sun-appserver-pe8.0|javax.enterprise.system.container.ejb|...|
javax.ejb.AccessLocalException: Client not authorized for this invocation.
at com.sun.ejb.containers.BaseContainer.preInvoke(BaseContainer.java:...
at com.sun.ejb.containers.EJBObjectInvocationHandler.invoke(...)
```

Solution

Check whether the EJB module (`.jar`) or web module (`.war`) is packaged in an application (`.ear`) and does not have role-mapping information in application level, Sun-specific, `sun-application.xml`. For any application (`.ear`), security role-mapping information must be specified in `sun-application.xml`. It is acceptable to have both module-level XML and application-level XML.

Authentication is Not Working With the Solaris Realm

Check whether the installation and server startup was performed as a local user, instead of as the root user. Always start the Application Server as the root user, because the Solaris realm works only with the root user. It was not designed to work with any other local user. Note also that role mapping can happen on the local user.

Mutual Authentication Not Working With the Application Client

Description

This failure can occur when the keystore and truststore properties are not set properly.

Solution

Set the following properties on the JVM:

```
javax.net.ssl.keyStore=  
<keystore-file-path\>;javax.net.ssl.trustStore=<truststore-file-path\>
```

To use the application client, set the environment variable `VMARGS` to the following value:

```
-Djavax.net.ssl.keyStore=  
${admin.domain.dir}/${admin.domain}/config/keystore.jks  
-Djavax.net.ssl.trustStore=  
${admin.domain.dir}/${admin.domain}/config/cacerts.jks
```


Frequently Asked Questions

This section covers some common questions asked about the Application Server:

- [“What Happens When No Server Side Realm is Configured?”](#) on page 47
- [“Can I Use a PKCS12 Certificate for My Client Certificate?”](#) on page 47
- [“Can I See the TLS/SSL Handshake Information for an SSL Client?”](#) on page 48
- [“Can I Change the Keystore Password?”](#) on page 48
- [“How Do I Maintain a Session in JAX-RPC?”](#) on page 48
- [“How Do I Access the Naming Service From a Standalone Java Client?”](#) on page 49
- [“Are RMI-IIOP Stubs Needed to Access Remote EJBs?”](#) on page 50
- [“How Do I Change the Log Level for an Application Logger?”](#) on page 51

What Happens When No Server Side Realm is Configured?

When the application is configured (within XML files), but no server side realm is configured, the application is authenticated in the default realm. No error is thrown that indicates “No such realm.”

Can I Use a PKCS12 Certificate for My Client Certificate?

Is there a way to use my PKCS12 certificate for an SSL the application client or standalone client during mutual authentication?

No. You cannot use a PKCS12 certificate directly, but you can write your own client using the JSSE, which supports `storetype=PKCS12` (read only, no write to keystore).

Can I See the TLS/SSL Handshake Information for an SSL Client?

Yes. Set the Java debugging property on the JVM. To see the handshake information from the application client, append the following:

```
-Djavax.net.debug=ssl,handshake to the VMARGS variable.
```

Can I Change the Keystore Password?

Yes. Use the following J2SE properties to change the keystore password:

```
-Djavax.net.ssl.keyStorePassword=password  
-Djavax.net.ssl.trustStorePassword=password
```

Note that the keystore password must match the individual key passwords to perform operations on the keys, so you will need to change the keystore password with the property mentioned above and then change the password to each key to match that password.

How Do I Maintain a Session in JAX-RPC?

Clients cannot maintain sessions with JAX-RPC endpoints. There is a client and server aspect to sessions, and it is not obvious how to set this up.

The situation is that a client makes a call to the service, and the server responds and sets a cookie on the connection. From then on, the client sends back that same cookie with each call and the server can check it.

A JAX-RPC stub normally ignores the cookie that comes back. When the `SESSION_MAINTAIN_PROPERTY` is set to `true`, it sends back whatever cookie the server set on it.

On the server side, you need to add one field to your class and a method that sets it. The endpoint must implement `javax.xml.rpc.server.ServiceLifecycle`, and two methods must be added: `destroy()` (which can be empty) and `init(Object context)`.

Add a `ServletEndpointContext` object to the endpoint; for example `myServletEndpointContext`. The `init(Object context)` method can be set as follows:

```
myServletEndpointContext = (ServletEndpointContext) context;
```

From then on, the business methods can access to the `HttpSession` with `myServletEndpointContext.getSession()`. The first call to `getSession` creates the session, if one does not already exist.

With this model, any method the client calls can get the session, set session attributes, get values from it, and so on. From then on, the client will send back the same cookie information.

For more information on the `HttpSession` object, see <http://java.sun.com/j2ee/1.4/docs/api/javax/servlet/http/HttpSession.html>

How Do I Access the Naming Service From a Standalone Java Client?

▼ To access the naming service from an application client

- Steps**
1. **Include `appserv-rt.jar` in the `CLASSPATH` when starting the client Java VM.**
The JNDI bootstrapping machinery looks for a file called `jndi.properties`, which is located in `appserv-rt.jar`. This file contains all the bootstrapping properties for the Application Server's naming service. It is better to have these properties read from `appserv-rt.jar` than to hard-code them in either the client startup script or in the application code.
 2. **When accessing remote EJBs from a standalone client, it is not necessary to retrieve the client JAR from the deployment or to put it in the client JVM's `CLASSPATH`, because static RMI-IIOP stubs are not needed when using the Application Server naming service. This removes a step that was required in previous releases. (See ["Are RMI-IIOP Stubs Needed to Access Remote EJBs?"](#) on page 50 for more details).**

3. **Code the client to use the default constructor `InitialContext` that does not require an argument. For example:**

```
InitialContext ic = new InitialContext();
```

It is a common misconception that the client should be coded to explicitly reference the `CosNaming` service. `CosNaming` is only used for some kinds of Application Server objects, so doing this will not provide access to many of the kinds of resources you might need in the client such as JMS queues, connection factories, and so on. Furthermore, explicit use of `CosNaming` bypasses the Application Server's naming service code. This often means that the client cannot take advantage of desirable value-added behavior built in to the Application Server's naming service.

4. **Use the global JNDI name of the target resource when doing the lookup. `java:comp/env` cannot be used from standalone Java clients, because by definition such clients run outside of a J2EE container. The only client component in which `java:comp/env` can be used is in a J2EE Application Client.**
5. **If the client is running on a different host machine than the server instance, set the following system property when starting the Java VM:**

```
-Dorg.omg.CORBA.ORBInitialHost=hostname_of_target_server
```

This value defaults to `localhost` so it is only needed if the client and server instance are not colocated. For example:

```
java -Dorg.omg.CORBA.ORBInitialHost=server1 ... com.foo.MyMainClass
```

6. **By default, the client attempts to contact port 3700 to access the naming service in the server. Since 3700 is the default naming service port used by the Application Server, there is no additional port configuration needed in the client. In some cases, due to port conflicts, the server instance uses a different naming service port. The naming service port used by the server instance is listed in the `<iiop-listener id="orb-listener-1" port="3700">` element in `domain.xml`.**

To change the naming service port used by the client, set the following system property when starting the client Java VM:

```
-Dorg.omg.CORBA.ORBInitialPort=naming_port_of_target_server
```

Are RMI-IIOP Stubs Needed to Access Remote EJBs?

No. Unlike previous releases of the Application Server, the current version does not require static RMI-IIOP stubs at runtime.

Removing this requirement provides the following benefits:

- Faster deployment and redeployment time for applications containing remote EJBs or clients of remote EJBs
- Fewer runtime errors caused by stub CLASSPATH configuration problems

In addition, the Application Server achieves these benefits without significant impact on runtime performance, while maintaining full RMI-IIOP interoperability.

The only scenario where RMI-IIOP stubs are still required is for standalone clients that explicitly instantiate an `InitialContext` for the `CosNaming` naming service. This is *not* the recommended approach for using the naming service in the Application Server. However, to maintain compatibility for these kinds of standalone clients, there is a deployment-time option that forces the generation of RMI-IIOP stubs in a way that matches previous releases. To use it, set

```
--generateterminstubs=true
```

when deploying with `asadmin` or the Administration Console. The RMI-IIOP stubs are placed in the `client.jar` file, just as they were in previous releases.

How Do I Change the Log Level for an Application Logger?

Each application uses its own application logger to log messages. To configure the log level for a particular application, use one of two options:

- In the Admin GUI Log Level configuration page add a property with a property name representing the logger name, and the value representing one of seven log levels (`FINEST`, `FINER`, `FINE`, `CONFIG`, `INFO`, `WARNING`, `SEVERE`) or `OFF`.

For example, to change the log level of application logger named `com.X.Y` to `FINEST`, the property name would be `com.X.Y` and the property value would be `FINEST`. The change is reflected in the `domain.xml` file, and takes effect immediately. No Server restart is required.

- Directly add the property to the `<module-log-levels>` element in `domain.xml`, as shown below.

```
<module-log-levels admin="INFO" classloader="INFO" cmp="INFO"
cmp-container="INFO" configuration="INFO" connector="INFO"
corba="INFO" deployment="INFO" ejb-container="INFO" javamail="INFO"
jaxr="INFO" jaxrpc="INFO" jdo="INFO" jms="INFO" jta="INFO"
jts="INFO" mdb-container="INFO" naming="INFO" node-agent="INFO"
resource-adapter="INFO" root="INFO" saaj="INFO" security="INFO"
server="INFO" synchronization="INFO" util="INFO" verifier="INFO"
web-container="INFO">
  <property name="com.X.Y" value="FINEST" /\>
</module-log-levels>
```


Index

A

ACC log service, 17
admin password
 invalid, 33-34
 not known, 34-35
admin username
 invalid, 33-34
 not known, 34-35
Administration Console, cannot access, 31-32
application, cannot access, 33-35

C

can't recall admin user name or
 password, 34-35
cannot access Administration Console, 31-32
cannot access local server, 28-29
cannot access remote server, 29-31
cannot access server application, 33-35
Cannot find log files, 26-28

D

diagnosis of a problem, 12
documentation, 18
domain
 finding, 31
 restarting, 23-26

E

error messages, 14

F

forum, 19

H

hostname, finding, 31
HTTP server access logs, 27

I

invalid admin username or password, 33-34
investigation of a problem, 14

K

Knowledge Base, 18

L

local server, access fails, 28-29
log files, 15
log levels, 16
logs
 cannot find, 26-28

logs (Continued)
 directory, 27

M

message types, 27

N

node agent, restarting, 23-26

O

online forum, 19

P

pkginfo, 12
port number, how the default value can
 change, 29

R

remote server, access fails, 29-31
restarting server, 23-26
rpm, 12

S

server
 how default port number can change, 29
Server, log file, description of, 26
server
 log file, location of, 27
 will not start on windows, 35-36
server.log, 16
showrev, 12
start-domain command fails, 21-23
SunSolve, 18

T

thread dump, obtaining, 17

W

Windows
 log levels, 16
 server will not start, 35-36