Sun Java™ System

Application Server 7 Troubleshooting Guide

2004Q2 Update 1

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

Part No: 819 0599

Contents

About This Guide

This *Troubleshooting Guide* provides instructions for resolving common problems with the Sun Java System Application Server, Enterprise Edition. (Referred to hereafter as the “Application Server”.) Many issues that also affect the Standard Edition product are also included.

The following topics are addressed in the preface:

Who Should Use This Guide

This manual is intended to be used by anyone who is charged with solving failure-to-operate problems with the Application Server software. This can include technical support engineers, system administrators, network administrators, application server administrators, and developers who are responsible for restoring operational functionality.

.

NOTE	This document does not address performance tuning, optimization, or best practices for using the Application Server. It focuses on the problems that are most likely to occur during operation, and provides guidelines for restoring operation as quickly as possible.

This guide assumes the reader is familiar with the following:

- UNIX® operating system
- Client/server programming model
- Java programming language
- How to install enterprise-level software products
- How Sun Java System Application Server 7 works
- High-availability and clustering concepts
- Internet and World Wide Web

Using the Documentation

The Sun Java System Application Server Standard and Enterprise Edition manuals are available as online files in Portable Document Format (PDF) and Hypertext Markup Language (HTML).

The following table lists tasks and concepts described in the Sun Java System Application Server manuals. The manuals marked *(updated for 7 2004Q2)* have been updated for the Sun Java System Application Server Standard and Enterprise Edition 7 2004Q2 release. The manuals not marked in this way have not been updated since the version 7 Enterprise Edition release.

Table 1 Sun Java System Application Server Documentation Roadmap

For information about	See the following
(Updated for 7 2004Q2) Late-breaking information about the software and the documentation. Includes a comprehensive, table-based summary of supported hardware, operating system, JDK, and JDBC/RDBMS.	*Release Notes*
Sun Java System Application Server 7 overview, including the features available with each product edition.	*Product Overview*
Diagrams and descriptions of server architecture and the benefits of the Sun Java System Application Server architectural approach.	*Server Architecture*
(Updated for 7 2004Q2) How to get started with the Sun Java System Application Server product. Includes a sample application tutorial. There are two guides, one for Standard Edition and one for Enterprise Edition.	*Getting Started Guide*

Table 1 Sun Java System Application Server Documentation Roadmap *(Continued)*

For information about	See the following
(Updated for 7 2004Q2) Installing the Sun Java System Application Server Standard Edition and Enterprise Edition software and its components, such as sample applications and the Administration interface. For the Enterprise Edition software, instructions are provided for implementing the high-availability configuration.	*Installation Guide*
(Updated for 7 2004Q2) Evaluating your system needs and enterprise to ensure that you deploy Sun Java System Application Server in a manner that best suits your site. General issues and concerns that you must be aware of when deploying an application server are also discussed.	*System Deployment Guide*
Creating and implementing Java™ 2 Platform, Enterprise Edition (J2EE™ platform) applications intended to run on the Sun Java System Application Server that follow the open Java standards model for J2EE components such as servlets, Enterprise JavaBeans™ (EJBs™), and JavaServer Pages™ (JSPs™). Includes general information about application design, developer tools, security, assembly, deployment, debugging, and creating lifecycle modules. A comprehensive Sun Java System Application Server glossary is included.	*Developer's Guide*
(Updated for 7 2004Q2) Creating and implementing J2EE web applications that follow the Java™ Servlet and JavaServer Pages (JSP) specifications on the Sun Java System Application Server. Discusses web application programming concepts and tasks, and provides sample code, implementation tips, and reference material. Topics include results caching, JSP precompilation, session management, security, deployment, SHTML, and CGI.	*Developer's Guide to Web Applications*
(Updated for 7 2004Q2) Creating and implementing J2EE applications that follow the open Java standards model for enterprise beans on the Sun Java System Application Server. Discusses Enterprise JavaBeans (EJB) programming concepts and tasks, and provides sample code, implementation tips, and reference material. Topics include container-managed persistence, read-only beans, and the XML and DTD files associated with enterprise beans.	*Developer's Guide to Enterprise JavaBeans Technology*
(Updated for 7 2004Q2) Creating Application Client Container (ACC) clients that access J2EE applications on the Sun Java System Application Server.	*Developer's Guide to Clients*
Creating web services in the Sun Java System Application Server environment.	*Developer's Guide to Web Services*
(Updated for 7 2004Q2) Java™ Database Connectivity (JDBC™), transaction, Java Naming and Directory Interface™ (JNDI), Java™ Message Service (JMS), and JavaMail™ APIs.	*Developer's Guide to J2EE Services and APIs*
Creating custom NSAPI plug-ins.	*Developer's Guide to NSAPI*
(Updated for 7 2004Q2) Information and instructions on the configuration, management, and deployment of the Sun Java System Application Server subsystems and components, from both the Administration interface and the command-line interface. Topics include cluster management, the high-availability database, load balancing, and session persistence. A comprehensive Sun Java System Application Server glossary is included.	*Administration Guide*

Table 1 Sun Java System Application Server Documentation Roadmap *(Continued)*

For information about	See the following
(Updated for 7 2004Q2) Editing Sun Java System Application Server configuration files, such as the `server.xml` file.	*Administrator's Configuration File Reference*
Configuring and administering security for the Sun Java System Application Server operational environment. Includes information on general security, certificates, and SSL/TLS encryption. HTTP server-based security is also addressed.	*Administrator's Guide to Security*
Configuring and administering service provider implementation for J2EE™ Connector Architecture (CA) connectors for the Sun Java System Application Server. Topics include the Administration Tool, Pooling Monitor, deploying a JCA connector, and sample connectors and sample applications.	*J2EE CA Service Provider Implementation Administrator's Guide*
(Updated for 7 2004Q2) Migrating your applications to the new Sun Java System Application Server programming model, specifically from iPlanet Application Server 6.x and Sun ONE Application Server 7.0. Includes a sample migration.	*Migrating and Redeploying Server Applications Guide*
(Updated for 7 2004Q2) How and why to tune your Sun Java System Application Server to improve performance.	*Performance Tuning Guide*
(Updated for 7 2004Q2) Information on solving Sun Java System Application Server problems.	*Troubleshooting Guide*
(Updated for 7 2004Q2) Information on solving Sun Java System Application Server error messages.	*Error Message Reference*
(Updated for 7 2004Q2) Utility commands available with the Sun Java System Application Server; written in manpage style.	*Utility Reference Manual*
Using the Sun™ Java System Message Queue 3.5 software.	The Sun Java System Message Queue documentation at: `http://docs.sun.com/db?p=prod/s1.s1msgqu`

How This Guide Is Organized

This guide contains the following chapters and appendixes:

- Chapter 1, "Troubleshooting Approach"
- Chapter 2, "Installation and Uninstallation Problems"
- Chapter 3, "Startup and Login Problems"
- Chapter 4, "Runtime Problems"
- Chapter 5, "HADB Issues on Windows"

Documentation Conventions

This section describes the types of conventions used throughout this guide:

General Conventions

The following general conventions are used in this guide:

- **File and directory paths** are given in UNIX® format (with forward slashes separating directory names). For Windows versions, the directory paths are the same, except that backslashes are used to separate directories.
- **URLs** are given in the format:

  http:// *server.domain* / *path* / *file*.html

  In these URLs, *server* is the server name where applications are run; *domain* is your Internet domain name; *path* is the server’s directory structure; and *file* is an individual filename. Italic items in URLs are placeholders.
- **Font conventions** include:
  - The `monospace` font is used for sample code and code listings, API and language elements (such as function names and class names), file names, pathnames, directory names, and HTML tags.
  - *Italic* type is used for code variables.

- *Italic* type is also used for book titles, emphasis, variables and placeholders, and words used in the literal sense.
- **Bold** type is used as either a paragraph lead-in or to indicate words used in the literal sense.

- **Installation root directories** for most platforms are indicated by *install_dir* in this document. Exceptions are noted in "Conventions Referring to Directories" on page 20.

  By default, the location of *install_dir* on **most** platforms is:

  - Solaris and Linux file-based installations:

    user's home directory`/sun/appserver7`

  - Windows, all installations:

    system drive`:\Sun\AppServer7`

  For the platforms listed above, *default_config_dir* and *install_config_dir* are identical to *install_dir*. See "Conventions Referring to Directories" on page 20 for exceptions and additional information.

- **Instance root directories** are indicated by *instance_dir* in this document, which is an abbreviation for the following:

  default_config_dir`/domains/`*domain*`/`*instance*

- **UNIX-specific descriptions** throughout this manual apply to the Linux operating system as well, except where Linux is specifically mentioned.

Conventions Referring to Directories

By default, when using the Solaris package-based or Linux RPM-based installation, the application server files are spread across several root directories. This guide uses the following document conventions to correspond to the various default installation directories provided:

- *install_dir* refers to `/opt/SUNWappserver7`, which contains the static portion of the installation image. All utilities, executables, and libraries that make up the application server reside in this location.
- *default_config_dir* refers to `/var/opt/SUNWappserver7/domains`, which is the default location for any domains that are created.

- *install_config_dir* refers to `/etc/opt/SUNWappserver7/config`, which contains installation-wide configuration information such as licenses and the master list of administrative domains configured for this installation.

Contacting Sun

You might want to contact Sun Microsystems in order to:

- Give Us Feedback
- Obtain Training
- Contact Product Support

Give Us Feedback

If you have general feedback on the product or documentation, please send this to `appserver-feedback@sun.com`.

Obtain Training

Application Server training courses are available at:

`http://training.sun.com/US/catalog/enterprise/web_application.html/`

Visit this site often for new course availability on the Sun Java System Application Server.

Contact Product Support

If you have problems with your system, contact customer support using one of the following mechanisms:

- The online support web site at:

  `http://www.sun.com/supportraining/`

- The telephone dispatch number associated with your maintenance contract

Please have the following information available prior to contacting support. This helps to ensure that our support staff can best assist you in resolving problems:

- 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. Here are some of the commonly used commands:
  - **Solaris:** `pkginfo, showrev`
  - **Linux:** `rpm`
  - **All:** `asadmin version --verbose`
- Detailed steps on the methods you have used to reproduce the problem
- Any error logs or core dumps
- Configuration files such as:
  - *instance_dir*`/config/server.xml`
  - a web application's `web.xml` file, when a web application is involved in the problem
- For an application, whether the problem appears when it is running in a cluster or standalone

Chapter 1

Troubleshooting Approach

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

The following sections are contained in this chapter:

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, Sun ONE 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 help you classify the problem, so you can more easily search for a solution.

NOTE Operating system utilities such as `pkginfo` and `showrev` on Solaris and `rpm` on Linux are helpful in gathering system information.

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 Sun ONE Application Server? On the same machine or not?
7. How is the Sun ONE Application Server connected to the directory server?
8. Are Sun ONE 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. Are the interoperating component versions in compliance with the compatibility matrix specified in the release notes?

After gathering this information, you will want to do the following:

- Collect web server error and access log data (web server instance-specific). Refer to "Server Logs" on page 125 for information on specific logs.
- Collect any Sun ONE Application Server stack traces.

NOTE 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 the user may have already taken in attempting to resolve the problem.

Seeking a Solution

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

The following topics are addressed in this section:

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:

- Information—These messages mark normal completion of particular 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 status to be reported as Failed. Error messages generally provide detailed information about the nature and the cause of the problem that occurred.

Error Messages

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

- In some cases, the message is very clear about what is wrong and what needs to be done to fix it. For example, you start up a domain using the asadmin start-domain command, then inadvertently issue the same command a few moments later, after the domain has been started. You get the following message:

  ```
  userD:\Sun\studio5_se\appserver7\bin>asadmin start-domain
  Domain already started : domain1
  Domain domain1 Started.
  ```

  In this case, the message gives clear guidance and the problem should be relatively easy to fix or can be disregarded.

- Sometimes an error message gives only general information on 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.iplanet.ias.config.ConfigException: Failed to load deployment descriptor for:
JAXBProjectStudio cause: java.io.FileNotFoundException:
```

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

- Some error messages are either not helpful, or give you no 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/appserver7/domains/domain1/server1/applications/j2ee-modules/Su
pplierServiceClient1_1/SupplierServiceClient1_SOAP.html (Overlapped I/O operation
is in progress.) status=1:5
```

  In this case, you have very little information to go on. It is especially important that you identify the exact situation that provoked 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 Messages Reference* at the following location:

http://docs.sun.com/db/coll/ApplicationServer7_04q2

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 Sun Java System Application Server has many log levels that can be set in the Administration interface (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.

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 Admin Server log file (`server.log`) and to tightly integrate with Solaris. The log levels ALERT and FATAL are specific to Application Server and are not implemented in the JDK1.4 logging API.

Log Options

The Administration interface 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.)

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 Sun ONE 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 to Clients* for more information.

Obtaining a Thread Dump

This section explains how to obtain a thread dump for Application Server 7. By default, the server dumps a core file and restarts with the `-Xrs java-option` flag in the `server.xml` file.

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 server instance.
3. Run the following command on the application server instance:

   `kill -3` *pid*

   The `kill` command will redirect the thread dump 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 if you need to seek more details, explanations, or examples.

Additional resources for researching an existing problem are available within Sun Microsystems:

- Sun Microsystems call center database
- Sun Microsystems bugtraq database

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 here:

```
http://docs.sun.com/db/coll/ApplicationServer7_04q2
```

A description of the Application Server manuals can be found in "Using the Documentation" on page 16.

Search the Knowledge Base

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

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 Free SunONE Articles.
4. Click Next.
5. Enter your search criteria.
6. Click Go.

Search or Participate in the Online Forum

You can browse directly in any of the online forums, or log in and register to start posting your own messages. The Application Server online forum is here:

```
http://swforum.sun.com/jive/index.jsp?cat=7
```

Contact Support

When necessary, gather together the information you have acquired and Contact Product Support, as described on page 21.

Chapter 2

Installation and Uninstallation Problems

The high-availability components of the Sun Java System Application Server, Enterprise Edition include the HADB, the HADB Management Client, and the load balancer plug-in. During installation, these components can be installed with the rest of the Application Server components, or separately. The load balancer plug-in is *usually* installed separately from the Application Server components.

This chapter addresses problems that you may encounter while performing installation or uninstallation of the Sun Java System Application Server, Enterprise Edition or its components or plug-ins.

The following sections are contained in this chapter:

Install/Uninstall Logs

This section describes the log files that are relevant when installing and uninstalling the Sun ONE Application Server. For a description of the format and the kinds of messages that can appear in a log file, see “Evaluate Messages” on page 25 and “Examine Log Files” on page 27.

The following Application Server logs can be useful for troubleshooting problems you may have with installation or uninstallation:

```
/var/sadm/install/logs/Sun_ONE_Application_Server_install.log
/var/sadm/install/logs/Sun_ONE_Application_Server_uninstall.log
```

Use the following log for troubleshooting problems with the `clsetup` command:
`/var/tmp/clsetup.log`

In addition to these log files, low-level installation and uninstallation log files are created at these locations:

```
/var/sadm/install/logs/Sun_ONE_Application_Server_install.<timestamp>
/var/sadm/install/logs/Sun_ONE_Application_Server_uninstall.<timestamp>
```

The following logs are associated with the high-availability components:

- Web server errors, including load balancer error messages, are written into the web server `error.log`.
- Application server messages, including deployment errors, are logged in the respective instance server's server.log file (the default location is `/var/opt/SUNWappserver7/domains/domain1/server1/logs`).
- Admin-server messages are logged in the admin-server's server.log file (the default location is `/var/opt/SUNWappserver7/domains/domain1/admin-server/logs`
- Database creation errors are written to `/var/sadm/install/logs/clsetup.log`.

- Initial cluster setup errors are written to `/var/tmp/clsetup.log`
- Cluster administration errors are written to `/var/tmp/cladmin.log`

Some guidelines on using the logs:

- Set the value of the `require-monitor-data` property to true in the `loadbalancer.xml` file in order to see monitoring details in the log.
- The `UnhealthyInstances` messages that appear in the log should be particularly helpful in troubleshooting load balancer problems.
- The `cladmin.log` file may be useful in troubleshooting cluster administration.
- The `clsetup.log` file may be helpful in finding out what went wrong during installation when you establish a new cluster.
- The HADB history files, described in “Examining the HADB history files” on page 149.

Can’t install remotely using the graphical interface

On UNIX, if you are installing the Application Server software remotely using the graphical interface, you must enable the display configuration on the machine where you are installing the product.

Solution

Set the DISPLAY environment variable to contain the name of the server and domain, using this format:

host_name`.`*domain_name*`.com:0.0`

Then run the following command on the remote client:

```
xhost +
```

Setup failure during Linux install

When an unsupported JDK is present, choosing “"upgrade JDK" or "Reuse existing JDK" can make the `setup` program fail while installing `SUNWicu`. Installing the RPM with `rpm -ihv` manually then gives a segmentation fault.

Solutution

Install the J2SE platform first, and then retry the Application Server installation.

Pre-existing JDK prevents installation, even after it has been removed

This error typically occurs on Solaris. The Application Server installer complains about a pre-existing JDK even though you removed the `/usr/j2se` directory. Installation does not proceed.

Explanation

The JDK can be installed in two different ways:

- As packages
- or as a loosely file based installation.

Whenthe JDK is installed as packages it always resides under `/usr/j2se`. It is then not enough to remove the directory. The packages have to be removed as well.

To find out if your system has any JDK packages, the Application Server executes the following commands as the root user:

```
pkginfo SUNWj3dev
pkginfo SUNWj3rt
pkginfo SUNWj3man
pkginfo SUNWj3dmo
```

If you receive a description of the queried package, this package is installed on your system.

Solution

Remove the JDK packages that installer checks for by executing the following commands as the root user:

```
pkgrm SUNWj3dev
pkgrm SUNWj3rt
pkgrm SUNWj3man
pkgrm SUNWj3dmo
```

Install or upgrade of J2SE fails

The following types of errors may occur if you attempt to upgrade your J2SE during installation:

- Incompatible J2SE version---cannot upgrade
- Failure to install J2SE

Incompatible J2SE version---cannot upgrade

This message occurs when the Solaris J2SE packages do not reside on the machine where you are peforming the installation, or when the J2SE version is not greater than or equal to version 1.3 and less than version 1.4.1_03.

NOTE The installation program can only upgrade a package-based J2SE installation, not a file-based J2SE installation.

Solution 1

Verify that the following packages are present using the `pkginfo -i -l` command:

```
SUNWj3rt
SUNWj3dev
SUNWj3man
SUNWj3dmo
```

Solution 2

Fixing the Solaris packages or completely removing the Solaris packages (if they are not used by any other application programs) using the `pkgrm` command.

You can then install the J2SE component using the installation program by selecting the Install Java 2 SDK (1.4.2_04) option in the Java Configuration panel.

Failure to install J2SE

The install log file reports that J2SE could not be installed. This error can occur when the `J2SE` directory (`/usr/j2se`, by default) is not writable by the user performing the installation.

Solution

Make the J2SE directory writable.

Can’t reinstall the server

If installation and uninstallation are performed according to the documented instructions and they complete normally, you will be able to reinstall the server with no problems. However, if you have used another method to remove the Application Server files, or if there as been a failure during installation or uninstallation, the system might be in an inconsistent state, leaving behind files or processes specific to theApplication Server in the `/var/sadm/install/productregistry` file. These leftover files and processes will provoke an error message similar to the following on a subsequent installation:

```
UnsatisfiedLinkError: Can't load library: libinstallCore.so
```

You will need to clean up these files or processes before attempting a new installation.

Solution: Clean up leftover files and processes

1. Log in as root.
2. Navigate to your installation directory and check the content of the `/var/sadm/install/productregistry` file for installed packages, that is, files having the SUNW string. For example:

   ```
   cat /var/sadm/install/productregistry | grep SUNW
   ```

3. Run `pkgrm` for the SUNW packages that were found in the product registry. For example:

   ```
   pkgrm SUNWasaco
   ```

4. Remove the following files, if they are present:

   ```
   /tmp/setupSDKNative
   ```

   ```
   /tmp/SolarisNativeToolkit_3.0_1
   ```

5. After the packages have been removed, use the `prodreg` registry editor to remove the Application Server-specific entries.

6. At the command line, kill all `appservd` processes that may be running by typing the following:

   ```
   ps -ef | grep appservd
   ```

   ```
   pkill appservd
   ```

7. Remove all remaining files under the Application Server installation directories. Refer to "Conventions Referring to Directories" on page 20 for further information and bundled and unbundled structures.

8. If present, remove the following log file:

   ```
   /var/sadm/install/logs/Sun_ONE_Application_Server_install.log
   ```

   (This step is helpful because the installation process appends information to this file, if it exists.)

9. Remove the Application Server directories.

   - Installation directory—by default, `/opt/SUNWappserver7`
   - Configuration directory—by default, `/etc/opt/SUNWappserver7`
   - Default domains directory—by default, `/var/opt/SUNWappserver7`

Silent installation is not working correctly

Consider the following:

- Is the silent installation configuration file correct?

Is the silent installation configuration file correct?

To run a silent installation, you must have created a silent installation configuration file by running a standard installation using the `savestate` option as described n the *Sun Java System Application Server Installation Guide.*

```
./setup -savestate
```

In tailoring the file for your silent installation, if you have introduced any errors in the configuration file, for example mistyping a variable name, the silent installation may not run.

Solution

Verify that the silent installation configuration file is correct and that you have not introduced any errors that may invalidate the file.

Uninstallation failure needs cleanup

If an uninstallation fails, you may need to clean up some leftover files or processes before attempting a new installation.

Solution

Follow the instructions in "Solution: Clean up leftover files and processes" on page 36.

Can't install the load balancer plug-in

Consider the following possibilities:

- Is your web server installed?
- Is there a previously installed load balancer or reverse proxy plug-in on your system?
- Has the load balancer plug-in already been installed?
- Are the configuration files correct?

Is your web server installed?

Before you can install the load balancer plug-in, you must have the web server already installed (Sun ONE Web Server 6.0, SP6 and above, or Apache Web Server 1.3.27). The web server is not required for the other Enterprise Edition components, just for the load balancer plug-in.

Solution

Install the web server before installing the load balancer plug-in.

Is there a previously installed load balancer or reverse proxy plug-in on your system?

The Sun Java System Application Server 7, Enterprise Edition requires that any existing load balancer or reverse proxy plug-in that exists on your system be removed before installing the load balancer plug-in.

Solution

Remove the existing plug-in using the uninstallation program. On a clean system, the following message should display if you try to access the plug-in:

```
ERROR: information for “SUNWaspx” was not found.
```

Has the load balancer plug-in already been installed?

If the load balancer plug-in component is disabled or grayed out on the Component Selection page, the correct version is already installed.

Are the configuration files correct?

The installation program checks to see if the appropriate configuration files for the load balancer plug-in are found in the location you specify.

For the Web Server plug-in, the following files are searched:

<*install_dir*>/config/magnus.conf
<*install_dir*>/config/obj.conf

For the Apache Web Server plug-in, this file is searched:

<*install_dir*>/conf/httpd.conf

Solution

Specify the correction location.

Shared memory creation failed

This error occurs while running `hadbm create` or `clsetup` (which calls `hadbm create`). When the HADB server processes are booted for the first time on each machine in the HADB configuration, they create the shared memory segments which constitute the database.

The typical message in this case is:

```
Failed to create shared memory
```

This message indicates that the `hadbm create` command could not allocate the shared memory to the database segments.

If you see this error in the history file, consider the following:

- Have you configured shared memory?
- Is there an error in your /etc/system file?
- Did you reboot the machine after configuring shared memory?
- Are there any other processes consuming shared memory?
- Are old AS or HADB installations occupying shared memory and semaphores?

Have you configured shared memory?

Shared memory must be configured for the HADB host machines before you can work with the HADB.

Solution

Configure shared memory by following the instructions in the Configuring Shared Memory and Semaphores section in the Preparing for HADB Setup chapter of the *Sun Java System Application Server Installation Guide*. For detailed information on the settings to use, consult the Performance and Tuning Guide, “Tuning for High Availability:Tuning HADB:Operating System Configuration”.

Is there an error in your /etc/system file?

You may have made a mistake or a typing error when you configured shared memory for the HADB.

Solution

Verify that you have followed the instructions in the Configuring Shared Memory and Semaphores section in the Preparing for HADB Setup chapter of the *Sun Java System Application Server Installation Guide*. Correct any typing error.

Did you reboot the machine after configuring shared memory?

The shared memory changes in the `/etc/system` file will not take affect until you have rebooted the machine.

Solution

Reboot the machine.

Are there any other processes consuming shared memory?

This situation can occur when there are too many AS instances and HADB nodes on the same machine.

Solution

Move some of the processes to another machine.

Are old AS or HADB installations occupying shared memory and semaphores?

Solution

Use the `ipcs` command to check the shared memory. If you find that the shared memory segments or semaphores are occupied unnecessarily, release them using `ipcrm -s` for semaphores and `ipcrm -m` for shared memory (in Unix systems).

clsetup is not working

The `clsetup` command is used to automate the process of setting up a cluster. After the Sun Java System Application Server 7, Enterprise Edition software and high-availability components are installed, this script uses three input files to set up a basic cluster. The most likely problems are errors in the input files (if they have been edited) and `clsetup` requirements not being met.

Consider the following possibilities:

- Was a previous clsetup terminated prematurely?
- Have you configured shared memory?
- Has remote communication been set up correctly?
- Under SSH, are the HADB and the Application Server co-located on the same machine?
- Are the application server and HADB installed in the same directories on each machine?
- Are all the Admin Servers on the application server instances in the cluster running?
- Are the input files on all instances in the cluster identical?

Was a previous clsetup terminated prematurely?

During `clsetup` , the HADB is created (a process that takes time and requires a bit of patience). Terminating `clsetup` during that process can leave HADB in an indeterminate state. This situation can produce a variety of a errors, including a `SessionStoreException` when creating database tables.

Solution

Ddelete HADB and rerun `clsetup`.

Have you configured shared memory?

Shared memory must be set up before you can use the `clsetup` command. See “Have you configured shared memory?” on page 40.

Has remote communication been set up correctly?

RSH or SSH must be set up before the `clsetup` command can be run.

To verify that remote communication has been established, `rsh` to each host in the cluster. The identity should be returned from the remote host. For example:

```
rsh computer99.zmtn.company.com uname -a
```

Instructions for setting up host communications are contained in the Preparing for HADB Setup chapter of the Sun Java System Application Server *Installation Guide.*

TIP After the SSH environment is set up, the very first time the SSH is invoked, you need to add the node machine name to the `known_hosts` file. Type the following: `ssh` *machine_name*

SSH will prompt you to add the *machine_name* to the `known_hosts` file by asking a yes/no question. Answer yes.

Solution

If the verification does not work, remote communication for the cluster has not been set up correctly. make sure that if you are using SSH Make sure that the `scp` and `ssh` binaries or that softlinks to them exist in `/usr/bin`. For further instructions, see the Setting Up Host Communication section of the *Sun Java System Application Server Installation Guide.*

Under SSH, are the HADB and the Application Server co-located on the same machine?

If you are co-locating the HADB and the Application Server on the same machine using SSH, a `known_hosts` file must exist under the `/.ssh` directory. That file is necessary for the HADB management client to communicate with the HADB nodes. Since hadbm does not permit use of `localhost`, you must use the acutal host name, instead.

Solution

If the `known_hosts` file is not under the `/.ssh` directory, run the `ssh` *hostname* command and answer `yes` to the prompt.

Are the application server and HADB installed in the same directories on each machine?

The `clsetup` program can not work when the files are installed in different directories on different machines.

Solution:

Reinstall the Application Server and HADB in the same directories on each machine."

Are all the Admin Servers on the application server instances in the cluster running?

Before running the `clsetup` command, all the Admin Servers in the cluster must be running.

Are the input files on all instances in the cluster identical?

The `clsetup` command is not designed to set up each instance with different values. For example, this command cannot create a JDBC connection with different settings for each instance.

Solution

Verify that the input files are identical on all instances in the cluster.

HADB database creation fails

The error occurs when using `clsetup` to start the database. The typical message in this case is:

```
failed to start database : HADB Database creation failed
```

To determine the cause of the problem, inspect the `/var/tmp/clsetup.log` file. Some possible errors are:

- No available memory

No available memory

Insufficient memory is available to create the database.

Solution 1

This problem can occur when changes are made to `/etc/system` and the `init 6` command is given to reset the system. The following error message occurs in the database log file:

```
System aborted with message:
  'Could not create shared DictCache segment'
...Shared memory get segment failed'
```

To avoid this problem, do `sync;sync` as root user and then do `reboot` instead of `init 6`.

Solution 2

This error can occur when insufficient swap space has been allocated. Review the documentation on shared memory requirements in the Preparing for HADB Setup chapter of the *Sun Java System Application Server Installation Guide*.

Specified hosts are not reachable

This could happen when you run `clsetup` to configure the cluster. You might see errors similar to the following in the log file:

```
CREATING HADB DATABASE...

/opt/SUNWhadb/4.2.2-17/bin/hadbm create
--installpath=/opt/SUNWhadb/4.2.2-17
--configpath=/etc/opt/SUNWhadb/dbdef --historypath=/var/tmp
--devicepath=/opt/SUNWhadb/4 --datadevices=1 --portbase=15200
--spares=0 --inetd=false --inetdsetupdir=/tmp --devicesize=512
--dbpassword=password --hosts=eas-v880-1,eas-v880-1 hadb

hadbm:Error 22024: Specified hosts are not reachable: [ eas-v880-1 ]

         HADB Database creation failed.
```

Solution

See “hadbm command fails: host unreachable.” on page 136.

Too few semaphores

The history file contains the following entry:

```
No space left on device
```

This can be caused when the number of semaphores is too low. Since the semaphores are provided as a global resource by the operating system, the configuration depends on all processes running on the host, not only the HADB. This can occur either while starting the HADB, or during runtime.

Solution

Configure the semaphore settings by editing the `/etc/system` file. Instructions and guidelines are contained in the Configuring Shared Memory and Semaphores section of the Preparing for HADB Setup chapter of the *Sun Java System Application Server Installation Guide.*

hadbm create Fails

hadbm create fails with the folowing error message:

`Node-`*`x`* `NSUP` *`timestamp`* `HADB-S-00240: Illegal node number`

The likely cause for this error is that another process is occupying the port that the NSUP process on node *X* is trying to open.

To resolve this issue, check for processes running on this machine which may have taken the port that is required by the NSUP process.

HADB Database Nodes Cannot be Reached

If the HADB database nodes cannot be reached and the database does not function, check whether dynamic IP addresses (DHCP) are used for hosts used in `hadbm createdomain` (or in other hadbm commands).

Hosts using DHCP are not supported by HADB.

HADB Creation Failures on Windows platform

The following issues might show up when running HADB 4.4 on Windows.

- hadbm createdomain will hang if managemnt agents can't reach each other with multicast.

  If different machines in the domain is connected to a switch which does not forward multicast messages (to multicast addresses), the `hadbm createdomain` function will never terminate.

  Solution: Configure your network infrastructure for multicast messages.

- Unexpected behaviour when two management domains share HADB nodes using the same port number. (See bug ID 6155745 in *Sun Java System Application 7 2004 Q2 Update 1 Release Notes*).

  The unexpected behavior could be node restarts, network partitions or reconnects with messages "Network Partition: *** Reconnect detected ***", written in HADB history files as well as on the HADB host terminals.

  In such cases, messages from nodes belonging to one database instance could be delivered to nodes belonging to the other database instance. This will lead to different problems, e.g. false network partitions and reconnects of partitions detected.

  Solution: If management domains share HADB hosts, ensure that the nodes on the common host do not use the same port number.

- hadbm create/addnodes hangs.

  hadbm create will give error when using a host with both single and multiple nets.

  Scenario: A host has multiple network interfaces. The user issues commands, `hadbm create`/`hadbm addnodes`.

  Solution: If a host has multiple network interfaces, specify the network interface to be used by HADB when issuing the commands, `hadbm create`/`hadbm addnodes`. If the hostname is used, the first interface registered on the host will be used, and there is no guarantee that the HADB nodes will be able to communicate.

Problems when running clsetup as non-root

If you want to run the `clsetup` command as a user other than root, you'll need to set up administration for non-root.

Solution

Follow the instructions in the Setting Up Administration for Non-Root section in the *Sun Java System Application Server Installation Guide.*

Can't test the ssh setting as root

In trying to test the SSH setting using the following command:

```
# ssh hostname date
```

the console prompts for the root password:

```
# root@hostname's password:
```

In Solaris 9, when you are using Sun verison of SSH software and running the HADB admin clients as root, the `sshd` configuration (`/etc/sshd_config`) on all machines in the cluster must have `PermitRootLogin` set to yes. Sun SSH does not permit root login by default; it is set to no.

Solution

1. Change `PermitRootLogin` on all machines to yes.
2. Restart `sshd`.

Can't get ssh to skip the login prompt

An error similar to the following occurs, suggesting that the `sshd` server is not running on the destination machine:

```
Secure connection to vortex-dev1 refused; reverting to insecure method.
Using rsh.  WARNING: Connection will not be encrypted.
Password:
```

You can set up your local environment to use the HADB commands from anywhere by setting the PATH variable after you have implemented SSH. You should not have to log in.

Solution

1. Verify that the SSH server is running by issuing the following command on the server machine:

   ```
   ps -e |grep sshd
   ```

2. If the SSH server is not running, start it as follow:

   ```
   /etc/init.d/sshd start
   ```

3. Check the `~<ssh-user>/.ssh/authorized_keys` file on each destination machine to ensure that all the public keys from all the machines are listed in that file.

4. For both the users home directory (`~<ssh-user>`) and the `.ssh` subdirectory, ensure that write permission is not granted for other or for group

For further information on setting up host communications for the HADB, refer to the Preparing for HADB Setup chapter of the *Sun Java System Application Server Installation Guide.*

NOTE If you are using a different version of SSH than the version described in the Sun Java System Application Server documentation, consult the documentation for that SSH version for instructions on setting up public key authentication.

Error configuring JMS Physical Destinations

The following error message occurs when you attempt to configure Physical Destinations for a JMS Service node:

```
[C4003]: Error occured on connection creation. - caught
java.net.ConnectException JMSService not available.
```

This error means that the app server instance could not connect to the JMS Service. Possible reasons for connection failure include :

- JMS Service (or App server instance) is not running .
- JMS Service is not startup enabled.
- JMS Service admin username and / or password is not valid.

Chapter 3

Startup and Login Problems

This chapter addresses common problems that can occur when the Application Server or Admin Server are starting, or when a user is logging in.

The following sections are contained in this chapter:

Can’t access server’s initial screen

When you visit the start page of the Application Server, the initial screen does not appear. Consider the following:

- Is the Application Server running?
- During installation, did the initial server startup run successfully?
- Is the server available locally?

- Was the server started at the expected port?
- Is your proxy setting causing a problem?
- Has an ungraceful shutdown occurred on a previously-running server?

Is the Application Server running?

Use one of the following commands to determine if the Admin Server has been started:

- On Unix:

  ```
  netstat -a | grep <admin-server port>
  ```

  Which produces a result like this:

  *.4848 *.* 0 0 24576 0 LISTEN

- On Solaris:

  ```
  ptree `pgrep appserver`
  ```

  which produces a list of all the processes and their directories. Check to see if `admin-server` appears in the output listing.

Solution

If the Application Server is not running, start the initially-configured administrative domain by running the following command:

```
asadmin start-domain --domain domain1
```

As the command completes, you should observe the following results:

```
asadmin start-domain --domain domain1
Instance domain1:admin-server started
Instance domain1:server1 started
Domain domain1 Started.
```

If other problems occur, you can use the following command to stop both the Admin Server as well as the Application Server instance of the initially-configured domain, domain1:

```
asadmin stop-domain --domain domain1
```

As the command completes, you should observe the following results:

```
asadmin stop-domain --domain domain1 --local
Instance domain1:server1 stopped
Domain domain1 Stopped.
```

Now restart the domain as explained above.

Syntax on the `asadmin` command is contained in the Application Server man pages and the *Sun Java System Application Server Administrator's Guide.*

During installation, did the initial server startup run successfully?

If the console window is still open, it should display a message like this:

```
Domain domain1 Started
```

where `domain1` is the name of the default domain. This indicates that the default domain was started successfully.

If you have already closed the console window, you can check for messages in the Application Server log file here:

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

If startup was successful, you should see a message similar to the following at the end of the log file:

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

Is the server available locally?

To verify that the server is running locally:

1. Log on to the machine where the server is running.
2. Access the local web page. For example

   `http://localhost:80/` (the default port)

Situation 1: If the start page does not appear on the local machine, it is most likely that the application server isn't running or didn't start normally.

Solution 1

In addition to checking the server logs for any errors during startup, check the following:

- “Is the Application Server running?” on page 52
- “Was the server started at the expected port?” on page 54.

Situation 2: If the start page appears locally but *not* on remote machines, there is a networking problem from the remote clients. For example, DNS might be set incorrectly (so the request is being sent to the wrong machine), the network configuration on the remote machine could be incorrect, a network router could be down, and so on.

Solution 2

This is probably not an Application Server issue. Check your network.

Was the server started at the expected port?

The server could be running at a different port number than the one you expect, 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:

1. Examine the server's configuration file here:

   domain_config_dir`/domains/domain1/server/config/server.xml`

2. Find the `http-listener` element.
3. Inspect the value of the port attribute.

Explanation of how the expected port number can change during installation—The server's default port number is 80, however, you can specify a different port number during installation. During installation, if the specified port number is already taken by another application when you start the server, the port number rolls forward to the next available number. For example, if a server was already running on the default port 80, the Application Server would be running on port number 81. If two servers were running, the port number would be 82, and so on.

If `http-listener` is running at a port that is in use, you may see a message similar to the following:

```
[21/Jan/2003:01:41:15] WARNING (10364): ADM0011: Could not
reregister HttpListener with DomainRegistry.
```

Application Server and HADB port assignments must not conflict with other port assignments on the same machine. Default and recommended port assignments are as follows:

- Sun Java™ System Message Queue: 7676
- IIOP: 3700
- HTTP server root: 80
- HTTP server non-root: 1024
- Admin server root: 4848
- HADB nodes: Each node uses five consecutive ports. If the default portbase (`15200`) is used, node 0 uses `15200` through `15204`, node 1 uses `15220` through `15224`, and so on.

Solution 1

Kill any other process that is running under the same port, or change the port number of the http-listener as follows:

1. Open the Administration interface (*hostname*:*admin_port*).
2. Browse to the HTTP Server.
3. Browse to the HTTP Listeners.

   The default listener is `http-listener-1`.
4. Click that default listener and find the port number (default is 80).
5. Change it to any unused port.
6. Save the settings.

   You should no longer receive this warning.

Solution 2

Change to another port and be sure to enter the correct port number when invoking the server.

Is your proxy setting causing a problem?

You should be able to access the server directly from your local system (localhost) as follows (for the default port 80):

```
http://localhost:80/
```

You may not be able to access your local system if your browser connects to the web through a proxy. (A proxy is a program that looks like a direct web connection, but which is actually a separate program that makes that connection for you.)

A typical error message situation is:

```
The requested item could not be loaded by the proxy.

Netscape Proxy's network connection was refused by the server:
localhost:4848 The server may not be accepting connections or may
be busy. Try connecting again later
```

Solution

To solve this problem, do one of the following:

- Direct your browser to bypass the proxy server when accessing your local system. Check your 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:80/
  ```

  To find your hostname and domain:

  - Type *hostname* at the command prompt.

Has an ungraceful shutdown occurred on a previously-running server?

If a crash has occurred, the server could be in an inconsistent state.

Solution

Use the `asadmin stop-domain` command to stop the Application Server, then restart the server using `asadmin start-domain` command.

Refer to “Is the Application Server running?” on page 52 for guidelines.

Can't access the Admin Server.

The Admin Server provides the administration facilities for the Application Server (one Admin Server per domain). The Application, Server log file, at `domains/domain1/admin-server/logs/server.log`, may be helpful in determining the reason the Admin Server is not running.

If you cannot access the Admin Server, consider the following:

- Has the Admin Server been started?
- Are you the user who installed the Application Server?
- Is the Admin Server running at the expected port?

Has the Admin Server been started?

See "Is the Application Server running?" on page 52.

Are you the user who installed the Application Server?

When the `start-domain` or `stop-domain` command fails with the error:

```
Could not start the domain.
You don't have permission to access
<install_dir>/domains/domain1/admin-server/config
```

The error indicates that you are not logged on as the user who installed the system.

Solution

You have to start the domain's admin server using the same login name as the user who installed the app server. You can then start *other* server instances using AdminGUI, once the admin server has been started, but the admin server can only be started by the person who installed the server.

Is the Admin Server running at the expected port?

The default port number for the Admin Server is 4848. However, the server could be running at a different port number than the one you expect, either because it was intentionally installed there, or because another server was already running on the installation port when the server was started.

Solution

Refer to “Was the server started at the expected port?” on page 54 for guidelines on checking the port your Admin Server is actually running on.

Be sure to enter the correct port number when invoking the Admin Server.

Can’t access a server application.

If you are unable to access a particular application, find the application’s context root in the deployed application’s `application.xml` file in `domains/domain1/server1/applications/j2ee-apps`.

Then consider the following:

- Is the Application Server running, and is the application deployed?
- “Is the application enabled?” on page 59
- Was application deployment successful?

Is the Application Server running, and is the application deployed?

The server must be running before an application can be accessed.

Solution

Use the `asadmin` command to determine if the application server is running. If it is, and if the application is deployed, the following command will list the deployed applications and components:

```
asadmin list-components --user admin --password password server1
```

You can then look for your application in the listing, which will look something like this:

```
hello1 <application>
dukesbook <application>
There are no standalone WAR modules
There are no standalone EJB modules
There are no connector modules
```

For more information, see "Can't access server's initial screen" on page 51.

Is the application enabled?

Use the following command to see if the application is enabled:

```
asadmin show-component-status --user admin --password password
dukesbook
```

where dukesbook is the application (component) name.

Was application deployment successful?

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

Solution

To verify that deployment was successful, do the following:

1. Check *install_dir*/domains/domain1/server/server.log for Admin Server. You may see entries similar to the following:

```
[20/Jul/2003:11:41:41] INFO ( 1600): DPL5109: EJBC - START of EJBC for
[stateless-converter]

[20/Jul/2003:11:41:41] INFO ( 1600): CORE3282: stdout: Remote message: Processing
beans ....

[20/Jul/2003:11:41:42] INFO ( 1600): DPL5108: EJBC - Generated code for remote
home and EJBObject implementations for [stateless-converter]

[20/Jul/2003:11:41:42] INFO ( 1600): CORE3282: stdout: Remote message: Compiling
wrapper code ....

[20/Jul/2003:11:41:46] INFO ( 1600): CORE3282: stdout: Remote message: Compiling
RMI-IIOP code ....

[20/Jul/2003:11:41:55] INFO ( 1600): DPL5110: EJBC - END of EJBC for
[stateless-converter]
```

```
[20/Jul/2003:11:41:56] INFO ( 1600): Total Deployment Time: 17605 msec, Total EJB
Compiler Module Time: 14100 msec, Portion spent EJB Compiling: 80%

Breakdown of EJBC Module Time: Total Time for EJBC: 14100 msec, CMP Generation: 0
msec (0%), Java Compilation: 10 msec (0%), RMI Compilation: 13239 msec (93%),

[20/Jul/2003:11:41:56] INFO ( 1600): ADM1041:Sent the event to
instance:[ApplicationDeployEvent -- deploy stateless-converter]

[20/Jul/2003:11:42:03] INFO ( 1600): ADM1042:Status of event to instance:[success]
```

2. Check the file system hierarchy under your server (such as server1) and look for your new application directory under `j2ee-apps`. If it was a module you deployed, look under the `j2ee-modules` directory to see your new module directory.
3. Check the instance's `server.xml` file in the `/config` directory for the instance. Look for an entry similar to the following for your application or module:

```
<j2ee-application enabled="true"
location="/Sun/studio5_se/appserver7/domains/domain1/server1/applications/j2ee-ap
ps/stateless-converter_1" name="stateless-converter" virtual-servers="server1"/>
```

Forgot the user name or password.

See “Don’t know the admin username / password” on page 132.

Forgot the admin Server port number.

See “Don’t know the Admin Server port number” on page 133.

Server won’t start

Some possible startup failure scenarios include:

- File parsing failure: loadbalancer.xml not found
- Invalid password(s)
- Abnormal subprocess termination / core dump
- CGI error

File parsing failure: loadbalancer.xml not found

The following error occurs when the configuration file has an invalid pointer to a file:

```
LBConfigParser...: reports: ... Parsing of file : Failed ...
Message:The primary document entity could not be opened.
```

For example, when the path to load balancer is specified incorrectly, the remainer of the message looks like this:

`Id=/...`*`path`*`.../config/loadbalancer.xml`

This error indicates that the path to the load balancer is invalid. You need to specify the absolute path to `loadbalancer.xml` in the Web Server configuration file

Invalid password(s)

The server instance log reports that startup was unsuccessful because of incorrect security password(s).

Solution 1

If you typed the wrong password three times during startup as a local user, you'll need to reinitiate the startup process.

Solution 2

If the wrong password was provided by a remote GUI/CLI instance startup, the procedure must be modified to supply the correct password.

Solution 3

If the security attribute in the `init.conf` file was wrongly set "off" for a secured instance, then it needs to be manually corrected.

Solution 4

If the `password.conf` file is present in the `config` directory and it contains the wrong passwords, it should be manually corrected, or deleted to initiate on-line requests for passwords.

Abnormal subprocess termination / core dump

The following error messages occur when attempting to the start the server:

```
Could not start the instance: domain1:admin-server
server failed to start: abnormal subprocess termination
Could not start the instance: domain1:server1
server failed to start: abnormal subprocess termination
Could not start one or more instances in the domain : domain1
Could not start one or more domains
```

Subsequent attempts to start the server may be accompanied by a core dump like the following, with errors recored in the server.log file:

```
CORE1116: Application Server
INFO: CORE3016: daemon is running as super-user
Bad System Call - core dumped
```

This error can occur when the app server runs out of file descriptors.

Solution: Increase the number of file descriptors

On UNIX, you can use the `ulimit` command to determine the number of available file descriptors or to set limits on the system's available file descriptors. The `ulimit` command displays the limits for the current shell and its descendants.

For the `sh` shell, the `ulimit -a` command lists all the current resource limits. The `ulimit -n` command lists the maximum file descriptors plus 1.

Check the file descriptors on your solaris box with the following unix command:

```
ulimit -n
256
```

where 256 is the number of file descriptors returned by the command.

To successfully start the servers, the file desciptor count should be set to 1024. Edit the `/etc/system` file and add the following 2 lines:

```
set rlim_fd_max=4086
set rlim_fd_cur=1024
```

After adding these lines, reboot your system and check the file descriptor value again. It should now be as follows:

```
ulimit -n
1024
```

You should now be able to start the admin server and server1 instance.

CGI error

If the Application Server won't start, you may receive the following error:

```
[05/Aug/2002:01:12:12] SEVERE (21770): cgi_init reports:
HTTP4047: could not initialize CGI subsystem

(Cgistub path /export/home/sun/appserver7/appserv/lib/Cgistub),
err fork() failure [Not enough space]
```

This message indicates that the system requiresadditional resources.

Solution 1: Set adequate limits for file descriptors

For more details, see "Solution: Increase the number of file descriptors" on page 62.

Solution 2: Change kernel parameters

On UNIX, increase the system resources by modifying the `/etc/system` file to include the following entries:

```
set rlim_fd_max=4086
set rlim_fd_cur=1024
```

Reboot the system for the new kernel parameters to take effect.

After you have set the shell resources, the Application Server should start.

Load balancer won't start

This section covers cases in which the load balancer fails to initialize:

- Parser can't open loadbalancer.xml
- Identical instance names

Parser can't open loadbalancer.xml

The following error is reported:

```
CNFG1000: Parsing of file : ...
Message:The primary document entity could not be opened.
Id=<path>/loadbalancer.xml
...
lb.configurator: CNFG1014 : Error occured while
initializing Loadbalancer config Parser.
```

This error occurs when the configuration file does not specify an absolute path to the location of the Load Balancer plugin.

Solution

Ensure that the configuration file contains the correct, absolute value.

Identical instance names

A message similar to the following might appear in the load balancer log file when you try to start the load balancer:

```
lb.configurator: CNFG1008 : Multiple instances with the same name  :
are not allowed for the cluster cluster1
```

The most likely problem is that the load balancer configuration file, `loadbalancer.xml`, is not configured correctly.

Solution

Verify your `loadbalancer.xml` file and make sure that the instance name is unique.

Load balancer / web server won’t start—listener ports

This problem occurs when two instances have the same listener value—for example, if instance `foo` has listener value `bar:80` and instance `spam` has listener value `bar:80`.

The error messages that result look like this:

```
04/Sep/2003:13:01:08] warning ( 2938):  reports: lb.runtime: RNTM2029:
DaemonMonitor :http://hostname:81 : could be because of connection saturation

[04/Sep/2003:13:01:08] failure ( 2938): ServerInstance.cpp@265: reports:
lb.runtime:RNTM3002 : Failed to add listener multiple times: <instance name>

[04/Sep/2003:13:01:08] failure ( 2938): FailoverGroup.cpp@102:
reports:lb.failovermanager: FGRP1002: Instance <instance name> could not be added to
theFailoverGroup: cluster1

[04/Sep/2003:13:01:08] failure ( 2938): LBConfigurator.cpp@209:
reports:lb.cofigurator: CNFG1007 :ServerInstance <instance name> could not be added
onFailoverGroup cluster1

[04/Sep/2003:13:01:08] failure ( 2938): lbplugin.cpp@168:  reports:
lb.runtime:RNTM3004 : Failed to initialise load balancing subsystem
```

Solution

Make sure that the listener values for each instance are unique.

Restart operation fails

When an attempted restart fails, consider the following:

- SSL / TLS are enabled

SSL/TLS are enabled

Restart does not work if SSL / TLS are enabled.

Solution

Stop and then start the instance.

JMS failed to start.

The JMS failed to start.

TIP If your application does not require JMS functionality, disabling it can improve performance. To disable JMS, change the following settings in the `server.xml` file:

```
<jms-service port="7676" admin-user-name="admin"
admin-password="admin" init-timeout-in-seconds="30"
enabled="false">
```

- Are you attempting to start the instance as a non-root user?
- Do Solaris bundled and unbundled domains and instances have the same names?
- Do the imq logs have out of memory errors?

Are you attempting to start the instance as a non-root user?

When attempting to start the application server instance as a non-root user, the command fails and the following message is displayed:

```
Could not start the instance
```

In the log file for the instance (`server.log`), the following error message occurs:

```
JMS5035: Timed out after 30000 milliseconds while trying to
verify if the JMS service startup succeeded.
```

When started as root, the application server instance starts normally.

Solution

Verify the correct user owns the JMS broker instance by running the following command:

```
ls -l /opt/imq/var/instances/
```

For example, the broker files for server1 in domain1 will be in the `domain1_server1` directory. If this directory is owned by root, the ownership of the broker files must be changed to the appropriate user. For example, the following command changes the ownership of these files to the UNIX user `greg` in the `staff` group:

```
chown -R greg:staff /opt/imq/var/instance/domain1_server1
```

Unless this change is made, it is not possible for the Application Server to access these files, so the JMS broker (and ultimately the Application Server) cannot start.

Do Solaris bundled and unbundled domains and instances have the same names?

If your machine has the Solaris 9 bundled version of the Application Server software installed, and you then install the unbundled version of the Application Server, the Message Queue broker for these application server installations will be shared.

NOTE In general, only one or the other type of bundle should be used. It not necessary to install an unbundled Application Server if a bundled version is already available.

If you do not uniquely name your new domains and instances, you may receive the following errors when starting up the second instance with the same domain or instance name:

```
SEVERE: JMS5024: JMS service startup failed
SEVERE: CORE5071: An error occured during initialization
```

Solution

Give the (unbundled) domains and instances names that are different from the instances and domains in the bundled installation.

To avoid these errors, refer to the JMS Support chapter in the *Application Server Administrator's Guide* for guidance.

Do the imq logs have out of memory errors?

If the imq logs show out of memory errors, system tuning is necessary.

Solution

1. Upgrade system memory.
2. Decrease the app server's heapsize.
3. Add more swapspace.

NOTE Adding more swap space will increase the number of applications that can run, which may adversely affect system performance, as more swapping will occur.

For more information on optimizing your system, consult the Performance and Tuning Guide.

JMS failed to start.

Chapter 4

Runtime Problems

This chapter addresses problems that you may encounter while running the Application Server.

The following sections are contained in this chapter:

Runtime Logs

Refer to the logs and information in “Server Logs” on page 125 for information on using the logs to troubleshooting runtime problems.

Can’t stop a remote server instance

Using root to start an admin server instance for a remote non-root user can lead to diccifulties when attempting to stop the instance.

Explanation

When an admin server instance starts up there are three Unix processes which implement that instance - two watchdog processes and the admin instance process.

When you use root to start an admin instance the watchdog processes run as root, but the admin instance process performs a `setuid` to run as the instance owner. If the instance owner now attempts to stop the instance, the attempt will fail because the instance owner cannot stop the watchdog processes owned by root.

Then, even if root attempts to stop the instance remotely, the instance only *appears* to stop—the processes continues to run. For example:

```
asadmin stop-domain --user admin --password adminadmin --port 4848
DomainStoppedRemotely
```

```
pgrep 'appserv*'
19049 19048 19050
(3 processes)
```

Even though the asadmin client runs as root, the command is sent to the remote admin server instance, which is running as the non-root user, so the root watchdog processes can't be stopped. (The "DomainStoppedRemotely" message *really* means "Domain Requested to Stop"—it doesn't guarantee success.)

Solution

To actually stop the admin server instance, a local root user must intiate the stop request:

```
asadmin stop-instance --domain domain1 admin-server
Instance admin-server stopped

pgrep 'appserv*'
(no processes)
```

To avoidsuch problems in the future, it is best to not use root to start instance on behalf of other users. Instead, let those users start the instances directly.

Instance goes unused after restarting

An instance was down, and is now back up, but the access log shows that is not getting any requests.

This situation occurs when the Load Balancer has not been configured to regularly check the health of instances. When the instance was down, the Load Balancer marked it as unhealthy, and recorded that fact in the load balancer log. But now that the instance is up and running again, the Load Balancer doesn't know that its health has been restored.

Solution

Configure the Load Balancer to check the health of instance regularly by adding the health checker URL to loadbalancer.xml with a line like this:

```
<health-checker url="/pathToHealthChecker"
   interval-in-seconds="10" timeout-in-seconds="30" />
```

Note:
Consult the Performance and Tuning Guide for recommendations on settting `interval-in-seconds`. Checking too frequently degrades performance, but checking too seldom (or not all) creates periods in which the insance goes unused.

Can't access a web page

Browsers report the following error when accessing an application:

```
404 Not Found
The requested URL destination_URL was not found on this server.
```

This means that the web page they were attempting to access is not available at the location they specified. The error frequently occurs because the incorrect URL was specified, or it may be transient. But it could also indicate a problem with the server.

Solution

Check that the application is deployed and enabled. See Can’t access an application, next.

Can’t access an application

There are a number of possible reasons that you cannot access an application. A typical error message in this case is

Consider the following:

- Is the application deployed?
- Is your loadbalancer.xml file correct?
- Is the web server running?
- Has the correct port been specified for the web server?

Is the application deployed?

The most likely cause is that the application is not deployed. The `cladmin` command is used to deploy an application to all instances in your cluster. Refer to the *Sun Java System Application Server Administrator’s Guide* for deployment instructions. Otherwise, refer to the *Sun Java System Application Server Developer’s Guide* for non-cluster deployment guidelines.

Solution

If needed, redeploy the application.

Is your loadbalancer.xml file correct?

Check the web server log files to verify that the load balancer started. If it hasn’t, there may be errors about the `loadbalancer.xml` file written in the to the error log.

Consider the following:

- Is the web server running?
- Has the correct port been specified for the web server?

Is the web server running?

Verify that the web server has started.

Has the correct port been specified for the web server?

Determine the correct web server port number and verify that the correct port has been specified. Refer to "Is the Admin Server running at the expected port?" on page 58 for guidelines on determining the port number.

Server responds slowly after being idle

If the server takes a while to service a request after a long period of idleness, consider the following:

Does the log contain "Lost connection" messages?

Does the log contain "Lost connection" messages?

If the server log shows error messages of the form,

```
java.io.IOException:..HA Store: Lost connection to the server..
```

then server has to recreate the JDBC pool for HADB.

Solution: Change the timeout value

The default HADB connection timeout value is 1800 seconds. If the application server does not send any request over a JDBC connection during this period, HADB closes the connection, and the application server needs to re-establish it. To change the timeout value, use the `hadbm set SessionTimeout=` command.

Important Note:
Make sure the HADB connection time out is greater than the JDBC connection pool time out. If the JDBC connection time out is more than the HADB connection time out, the connection will be closed from the HADB side, but it will be there in appserver connection pool. So when the application tries to use the connection, the application server will have to re-create the connection, which incurs significant over head

Application suddenly goes away

Consider the following:

- Is the application you are using being quiesced by the load balancer?

Is the application you are using being quiesced by the load balancer?

When an application is being quiesced, you may experience loss of service when the application is disabled, until the application is re-enabled.

Requests are not succeeding

The following problems are addressed in this section:

- Is the load balancer timeout correct?
- Have you enabled the instances of the cluster?
- Are the system clocks synchronized?
- Is the AppServer communicating with HADB?

Is the load balancer timeout correct?

When configuring the `response-timeout-in-seconds` property in the `loadbalancer.xml` file, you must take into account the maximum timeouts for all the applications that are running. If the response timeout it is set to a very low value, numerous in-flight requests will fail because the load balancer will not wait long enough for the Application Server to respond to the request.

On the other hand, setting the response timeout to an inordinately large value will result in requests being queued to an instance that has stopped responding, resulting in numerous failed requests.

Solution

Set the `response-timeout-in-seconds` value to the maximum response time of all the applications.

Are the system clocks synchronized?

When a session is stored in HADB, it includes some time information, including the last time the session was accessed and the last time it was modified. If the clocks are not synchronized, then when an instance fails and another instance takes over (on another machine), that instance may think the session was expired when it was not, or worse yet, that the session was last accessed in the future!

NOTE In a non-colocated configuration, it is important to synchronize the clocks on that machines that are hosting HADB nodes. For more information, see the *Installation Guide* chapter, "Preparing for HADB Setup".

Solution

Verify that clocks are synchronized for all systems in the cluster.

Have you enabled the instances of the cluster?

Even if you start an application server instance and define it to be a part of the cluster, the instance will not receive requests from the load balancer until you enable the instance. Enabling makes the instance and active part of the cluster. The correct sequence of events for activating and deactivating an instance is:

1. Start the Application Server.
2. Create an Application Server instance.
3. Enable the instance
4. Disable the instance.
5. Stop the instance.
6. Start the instance (only if it has been stopped).
7. Enable the instance.

Is the AppServer communicating with HADB?

HADB may be created & running, but if the persistence store has not yet been created, created the Application Server won't be able to communicate with the HADB. This situation is accompanied by the following message:

```
WARNING (7715): ConnectionUtilgetConnectionsFromPool failed using
connection URL: <connection URL>
```

Solution

Create the session store in the HADB with a command like the following:

asadmin create-session-store --storeurl *connection URL* --storeuser haadmin --storepassword hapasswd --dbsystempassword super123

Server log: app persistence-type = memory

The server.log shows that the J2EE application is using memory persistence instead of High Availability, with a message like this:

```
Enabling no persistence for web module [Application.war]'s sessions:
persistence-type = [memory]
```

This situation occurs when the application server has not been configured to use HA.

Solution

Enable the availability service with a command like this:

```
asadmin set --user admin --password netscape --host localhost
--port 4848 serverName.availability-service.availabilityEnabled=true
```

Dynamic reconfiguration failed

The load balancer plug-in detects changes to its configuration by examining the time stamp of the `loadbalancer.xml` file. If a change has been made to the `loadbalancer.xml` file, the load balancer automatically reconfigures itself. The load balancer ensures that the modified configuration data is compliant with the DTD before overwriting the existing configuration.

If changes to the `loadbalancer.xml` file are not in the correct format, as specified by the `sun-loadbalancer_1_0.dtd` file, the reconfiguration fails and a failure notice is printed in the web server's error log files. The load balancer continues to use the old configuration in memory.

NOTE If the load balancer encounters a hard disk read error while attempting to reconfigure itself, it uses the configuration that is currently in memory, and a warning message is logged to the web server's error log file.

Solution 1

You may have to wait up to the amount time configured for the reload time interval before you see the change. Verify that the interval isn't longer than you expected.

Solution 2

Edit the `loadbalancer.xml` file as needed until it follows the correct format as specified in the DTD file.

Session Persistence Problems

The following problems are addressed in this section:

- The create-session-store command failed
- Configuring instance-level session persistence didn't work

The create-session-store command failed

Consider the following:

- Are the HADB and the application server instance on different sides of a firewall?

Are the HADB and the application server instance on different sides of a firewall?

The `asadmin create-session-store` command cannot run across firewalls. Therefore, for the `create-session-store` command to work, the application server instance and the HADB must be on the same side of a firewall.

The `create-session-store` command communicates with the HADB and not with the application server instance.

Solution

Locate the HADB and the application server instance on the same side of a firewall.

Configuring instance-level session persistence didn't work

The application-level session persistence configuration always takes precedence over instance-level session persistence configuration. Even if you change the instance-level session persistence configuration after an application has been deployed, the settings for the application still override the settings for the application server instance.

Session data seems to be corrupted

Session data may be corrupted if the system log reports errors under the following circumstances:

- During session persistence
- When the session state is read during session activation
- When the session state is read after session failover

If you determine that the data has been corrupted, there are three possible solution.

Solution

To bring the session store back to a consistent state, do the following:

1. Use the `asadmin clear-session-store` command to clear the session store.
2. If clearing the session store doesn't work, reinitialize the data space on all the nodes and clear the data in the HADB using the `hadbm clear` command.
3. If clearing the HADB doesn't work, delete and then recreate the database.

HTTP session failover is not working

The Sun Java System Application Server 7, Enterprise Edition includes the high-availability database (HADB) for storing session data. The HADB is not a general-purpose database but instead is an HttpSession store.

If HTTP session failover is not working correctly, consider the following:

Are the system clocks synchronized?

For HTTP session failover to work, the clocks of all the computers on which the application server instances in a cluster reside must be synchronized. (For more detail, see "Are the system clocks synchronized?" on page 75.)

Solution

Verify that clocks are synchronized for all systems in the cluster.

Do all objects bound to a distributed session implement the java.io.Serializable interface?

If an object does not implement the `java.io.Serializable` interface, it will not be persisted. No errors or warnings are produced, because the lack of persistence may well be the desired behavior. The remaining session objects are successfully persisted and will fail over.

NOTE If an object which is not serializable implements `java.io.Serializable`, an exception will be thrown in the log. In this case, the entire user session is not persisted. A failover will produce an empty session.

Solution

Make sure that every class in the session that is supposed to persist implements java.io.Serializable, as in this example:

```
public class MyClass implements java.io.Serializable
{
    ..
}
```

Is your web application distributable?

For a web application to be highly available, it should be distributable. An application is non-distributable if the `webapp` element of the `web.xml` file does not contain a distributable subelement.

For additional information, refer to the *Sun Java System Application Server Developer's Guide to Web Applications.*

Solution

Verify that the `webapp` element of the `web.xml` file contains a distributable subelement.

Is the persistence type set to ha?

The persistence type must be set to `ha` for session and SFSB failover to work. When you run the `clsetup` command, the persistence type is set to `ha` by default. If you do not use the `clsetup` command to set up your initial cluster, the persistence type is specified as `memory`, the default. The `memory` type offers no session persistence upon failover, while the failover capabilities offered by the `file` persistence type are intended for use only in development systems where failover capabilities are not strictly required.

Instructions for setting the persistence type are contained in the Session Persistence chapter of the *Sun Java System Application Server Administrator's Guide.*

Solution

Verify that the persistence type is set to `ha`. If it isn't, modify either the entire instance or your particular application to use the `ha` persistence type. For details, see the Session Persistence chapter of the Administration Guide.

An object is cloned instead of shared

When using `modified attribute` persistence scope, and a session fails over or is activated after being passivated, an object that is shared between two attributes comes back as two separate copies of the object instead of as a single shared object.

The situation occurs because you can not have one object referred to by two separate attributes when using `modified attribute`. The application server serializes and persists each attribute separately, so the shared object gets serialized twice, once for each attribute. When the objects get deserialized, they are now two separate objects.

Has a session store been created?

HTTP session and SFSB failover will not work until a session store has been created using the `asadmin create-session-store` command.

Instructions for creating a session store are contained in the Session Persistence chapter of the *Sun Java System Application Server Administrator's Guide*

Are all the machines in the cluster homogenous?

All Application Server instances in a cluster must have the same applications deployed to them. For these applications to take part in failover, they must have a consistent session persistence configuration and point to the same session store.

Any new instance that you add to a cluster must have the same version and same patch level as all existing instances in a cluster.

Has high availability been enabled?

HTTP session and SFSB failover will not work until high availability has been enabled using the `availability-enabled` attribute.

Solution

Set the `availability-enabled` attribute using the `asadmin` command.

Out of Memory and Stack Overflow errors

When using the Application Server to deploy web applications, out of memory errors and stack overflow errors are occurring., even though the volume of data is not large.

Memory: 2048M
CPU : 1 900Mhz Ultra SparcIII

Solution 1

Java VM runs out of memory because the web applications are quite demanding in terms of creating Java objects.

This can usually be solved by setting the VM `Xms` / `Xmx` and `Xss` parameters. If you have default values similar to these:

```
-Xmx256m - for initial app. server install
-Xss512k - default for 1.4.0 server VM
```

try something like:

```
-Xmx1024m -Xss1024k
```

Solution 2

Java VM runs out of memory because of bad web application design.

For instance, the application creates lots of references to Java objects and they are being cross referenced through the lifecycle of this application preventing the garbage collector from doing its job. With each additional user using the application, it consumes more and more heap space until it runs out.

There is nothing you can do here except for changing the application. You are most likely dealing with an application memory leak which can be traced using a profiling tool such as OptimizeIt.

Solution 3

The other thing you can try is to configure your Application Server with a later version of the J2SE platform, to take advantage of improvements in memory management and garbage collection.

Performance Problems

This section discusses the following issues:

Too much swapping in a colocated HADB system

When HADB is sharing a system with other processes, insufficient memory may lead to exceptionally high levels of swapping activity. When that situation arises, performance suffers and transactions may be lost.

If the system is running slowly, use the Unix `vmstat` command to check swapping levels, and look for this message in the HADB history files, where `M` is greater than `N`:

`Process blocked for .M. sec, max block time is .N. sec`

This message can occur when excessive swapping causes HADB restarts.

For additonal explanations and recommendations, see the Performance Tuning Guide section: “Tuning for High-Availability”, “Tuning HADB, “Memory Allocation”.

Indefinite looping in server.log after MDB failure

When invoking CMP from MDB (with a container managed transaction), there could be a problem if the CMP throws an EJBException, but a RuntimeException is not thrown in MDB and global transaction is set to true. The MDB container is then unaware of the exception. Because the message is part of the transaction, the JMS provider(Sun Java™ System Message Queue) will keep trying to deliver the message. In that case, the exception continues to get logged in `server.log` indefinitely.

Solution

To prevent indefinite looping with Message Queue, throw a runtime exception in the MDB so that the MDB container will be aware of it.

Performance suffers when using server-side SOAP message handlers

SOAP Message Handlers are commonly be used in commercial Web services for encryption and decryption before sending a message to its endpoint or setting new Headers (for example, setting a session id and handling attachments). However, they are known to reduce throughput.

Explanation

Bug #4733037 explains the problem . We have observed about a 50% reduction in throughput even with a completely empty MessageHandlers on the Server side.

Solution

For the sake of performance, it is best to minimize the use of SOAP message handlers on the server side, until we have a Service pack or performanec pack release to fix this issue.

Low Memory in MQ Broker

Sometimes while running the JMS related applications for long hours under stress, the MQ Broker goes to a low memory state.

Solution

The default Java heap size is `-Xms8m -Xmx128m -Xss128k`. Increasing the Broker heap size can help to remedy the issue.

1. Access the Admin Console at `http://<hostname>:<adminport>`.
2. Login to the Admin Console using the admin username and password.
3. Expand "App server instances" and select the instance where the JMS application is run.
4. Select JMS in the menu, expand it, and select Service.
5. In the right hand side we get the General and Properties lists.
6. Under the General list, select Start Arguments.
7. In the text box enter the heap size as: `-vmargs -Xms512m -vmargs -Xmx512m`(where 512m is an example of the heap size.
8. Click the SAVE button below the screen.

9. Select the server instance again on the left hand frame and click Apply Changes.
10. Start or restart the server instance if it is not running.
11. To verify that the broker has picked up the heap size set, inspect the broker log.

HADB Performance Problems

Performance will be affected when the transactions to HADB get delayed or aborted. This situation is generally caused by a shortage of system resources. Delays beyond five seconds causes the transactions to abort. Any node failures can also cause the active transaction on that node at the time of the crash to abort. Any double failures (failure of both mirrors) will make the HADB unavailable. The causes of the failures can generally be found in the HADB history files.

In your efforts to isolate the problem, consider the following:

- Is there a shortage of CPU or memory resources / too much swapping?
- Is There a Disk Contention?
- Is there a shortage of HADB data devices space?
- Is there a shortage of other HADB resources?
- Direct Disk I / O Mapping Failed Errors

Is there a shortage of CPU or memory resources/too much swapping?

Node restarts or double failures due to `Process blocked for` *x* `sec, max block time is 2.500000 sec`.

Here, x is the actual time duration the process was blocked and this is greater than 2.5 seconds.

The HADB Node Supervisor Process (NSUP / clu_nsup_srv) tracks the time elapsed since the last time it did some monitoring work. If that time duration exceeds a specified maximum (`2500 ms`, by default), NSUP concludes that it was blocked too long and restarts the node.

If NSUP is blocked for more than 2.5 seconds, it restarts the node. If mirror nodes are placed on the same host, the probability of the occurrence of double failure is high. Simultaneous occurrence of the blocking on the mirror hosts may also lead to double failures.

The situation is especially likely to arise when there are other processes, eg. in a colocated configuration, in the system that compete for CPU, or memory which produces extensive swapping and multiple page faults as processes are rescheduled. NSUP being blocked can also be caused by negative system clock adjustments.

Solution: Ensure that HADB nodes get enough system resources. Ensure also that the time syncronisation daemon does not make large (not higher than 2 seconds) jumps.

Is There a Disk Contention?

Disk contention can be give negative impact to user data read / writes to the disk devices, as well as to HADB writes to history files. Severe disk contention may delay or abort user transactions. Delay in history file writing may cause node restarts, in the worst case, lead to double failures.

The disk contention can be identified by monitoring the disk I/O from the operating sysytem, for the disks used for data, log devices and history files. This can also identified by the following statement in the history files:

```
HADB warning: Schedule of async <read,write> operation took ..
```

Write delays to the history file will be written to the HADB history files. This can be identified by,

```
NSUP0 BEWARE <timestamp>  Last flush took too long (x msecs).
```

This warning shows that disk I/O took too long. If the delay exceeds 10 seconds, the node supervisor restarts the trans process with the error message:

```
Child process trans0 10938 does not respond.

Child died - restarting nsup.

Psup::stop: stopping all processes.
```

This message indicates that a trans (clu_trans_srv) process has been too busy doing other things (eg., waiting to write to the history file) to reply to the node supervisor's request checking the heartbeat of the trans process. This causes the nsup to believe that the trans has died and then restart it.

This problem is observed especially in Red Hat Advanced Server 2.1 when multiple HADB nodes are placed on the same host and all the nodes use the same disk to place their devices.

Solution: Use one disk per node to place the devices used by that node. If the node has more than one data devices and the disk contention is observed, move one data device to another disk. The same applies to the history file as well.

Is there a shortage of HADB data devices space?

One possible reason for transaction failure is running out of data device space. If this situation occurs, HADB will write warnings to the history file, and abort the transaction which tried to insert and / or update data.

To find the messages, follow the instructions in "Examining the HADB history files" on page 149. Typical messages are:

```
HIGH LOAD: about to run out of device space, ...
```

```
HIGH LOAD: about to run out of device space on mirror node, ...
```

The general rule of thumb is that the data devices must have room for at least four times the volume of the user data. Please refer to the Tuning Guide for an explanation.

Solution 1

Increase the size of the data devices using the following command:

```
hadbm set TotalDataDevicePerNode=size
```

(See the Administrator's Guide)

This solution requires that there is space available on the physical disks which are used for the HADB data devices on all nodes.

HADBM will automatically re-start each node of the database.

Solution 2

Stop and clear the HADB, and create a new instance with more nodes and / or larger data devices and / or several data devices per node, see the Administrator's guide.

Unfortunately, using this solution will erase all persistent data.

See also, Bug ID 5097447 in *Sun Java System Application Server 7 2004Q2 Update 1 Release Notes*.

Is there a shortage of other HADB resources?

When an HADB node is started, it will allocate:

- several shared memory segments, of fixed size
- internal data structures of fixed size

If an HADB node runs out of resources it will delay and / or abort transactions. Resource usage information is shipped between mirror nodes, so that a node can delay or abort an operation which is likely to fail on its mirror node.

Transactions that are delayed repeatedly may time out, and return an error message to the client. If they do not time out, the situation will be visible to the client only as decreased performance in the period where the system is short on resources.

These problems frequently occur in “High Load” situations. For details, see “High Load Problems” on page 88.

Direct Disk I/O Mapping Failed Errors

When using Veritas file system on Solaris, you might get the message `WRN: Direct disk I/O mapping failed` in the HADB history files.

This message indicates that HADB is unable to turn on direct I/O for the data and log devices. Direct I/O is a performance enhancement that reduces the CPU cost of writing disk pages, and also causes less overhead of administering dirty data pages in the operating system.

To use direct I/O with Veritas, you should create the data and log devices on a file system that is mounted with the option `mincache=direct`. Note that this option will apply to all files created on the file system. For details, see the command `mount_vxfs(1M)`.

An alternative is to use the Veritas Quick I/O facility. In effect, this product makes it possible to perform raw I/O to file system files. For more information, see the Veritas document, *VERITAS File System™ 4.0 Administrator’s Guide for Solaris*.

NOTE This description is based on available documentation only.

Sun Database Technology Group has not tested these configurations.

High Load Problems

High load scenarios are recognizable by the following symptoms:

- User requests do not succeed.
- The database gives multiple timeout and 'transaction aborted' messages.
- Frequent “HIGH LOAD” warnings in the history file.
- Sporadic failures

When you suspect a high load scenario, consider the following:

- Is the Tuple Log out of space?
- Is the node-internal log full?
- Are there enough locks?
- Can you fix the problem by doing some performance tuning?

Note:
Frequently, these problems can be solved by making more CPU horsepower available, as described in "Improving CPU Utilization" on page 96.

. (For even more information, consult the Tuning Guide.)

Is the Tuple Log out of space?

The user operations (delete, insert, update) are logged in the tuple log and executed. There tuple log may fill up because:

- Execution slows due to CPU or disk I/O contention
- The mirror node is slow in receiving the log records ('log throw due to..'

  messages in the history files), which happens as a result of:

  - Network contention, so the log records don't reach the mirror node
  - CPU contention at the mirror node which keeps it from processing the received log records quickly enough.

Solution 1

Check CPU usage, as described in "Improving CPU Utilization" on page 96.

Solution 2

If CPU utilisation is not a problem, check the disk I/O. If the disk shows contention, avoid page faults when log records are being processed by increasing the data buffer size with `hadbm set DataBufferPoolSize=...`

Solution 3

Look for evidence of network contention, and resolve bottlenecks.

Solution 4

Increase the tuple log buffer using `hadbm set LogBufferSize=...`

See also, Bug ID 5097447 in *Sun Java System Application Server 7 2004Q2 Update 1 Release Notes*.

Is the node-internal log full?

Too many node-internal operations are scheduled but not processed due to CPU or disk I/O problems.

Solution 1

Check CPU usage, as described in "Solution 2: Improve CPU Utilization" on page 90.

Solution 2

If CPU utilization is not a problem, and there is sufficient memory, increase he `InternalLogbufferSize` using the `hadbm set InternalLogbufferSize=` command.

Are there enough locks?

Some extra symptoms that identify this condition are:

- Error code 2080 or 2096 delivered to the client.
- HIGH LOAD log messages thrown due to locks.
- `hadbm resourceinfo --locks` shows locks allocated, and all are in use., all the time

Solution 1: Increase the number of locks

Use `hadbm set NumberOfLocks=` to increase the number of locks.

Solution 2: Improve CPU Utilization

Check CPU usage, as described in "Improving CPU Utilization" on page 96.

Can you fix the problem by doing some performance tuning?

In most situations, reducing load or increasing the availability of resources will improve host performance. Some of the more common steps to take are:

- Run the nodes on hosts with better hardware characteristics (more internal memory, higher processor speed, more processors).
- Add physical disks and use several data devices, not more than one device on each physical disk.
- Add more nodes, on new hosts, and refragment the data to utilize the new nodes.
- Change configuration variables in order to allocate larger memory segments or internal data structures.

In addition, the following resources can be adjusted to improve “HIGH LOAD” problems, as described in the *Performance and Tuning Guide*:

Size of Database Buffer:	`hadbm attribute DataBufferPoolSize`
Size of Tuple Log Buffer:	`hadbm attribute LogBufferSize`
Size of Node Internal Log Buffer:	`hadbm attribute InternalLogBufferSize`
Number of Database Locks:	`hadbm attribute NumberOfLocks`

Client cannot connect to HADB

This problem is accompanied by a message in the history file:

```
HADB-E-11626: Error in IPC operations, iostat = 28: No space left on device
```

where:

- 11626 is an HADB error code, 'Error in IPC operations', which means that some Inter Process Communication operation failed.
- 'iostat = 28' means (on solaris) that the operating system set errno to ENOSPC, which again translates to 'No space left on device'.

The most likely explanation is that a `semget()` call failed (see the unix man pages). If HADB started successfully, and you get this message at runtime, it means that the host computer has too few semaphore “undo structures”. See the “Preparing for HADB Setup” chapter in the Installation guide for how to configure `semmnu` in `/etc/system`.

Solution

Stop the affected HADB node, reconfigure and reboot the affected host, re-start the HADB node. HADB will be available during the process.

Connection Queue Problems

The following problems may occur with connection queues:

- Full connection queue closes socket

Full connection queue closes socket

This problem typically occurs in “high load” scenarios. The `server.log` file shows the error: `SEVERE (17357): Connection queue full, closing socket`.

Solution: Increase configuration values

Increase the value for `ConnQueueSize` and `MaxKeepAliveConnections` in file `magnus.conf` under the `config` directory of your Web Server, for example:

- `MaxKeepAliveConnections 1024`
- `ConnQueueSize 1024`

Connection Pool Problems

The following problems may occur in relation to connection pools:

- Single sign-on requires larger connection pool
- Server: Unable to obtain connection from pool
- JDBC connection is not available
- Exception occurs while calling DataSource.getConnection()
- Exception occurs while executing against MSSQL
- IOException: Connection in invalid state

Single sign-on requires larger connection pool

When single sign-on (or session persistence) requires connections and the wait time is exceeded, the following error occurs:

```
Unable to get connection - Wait-Time has expired
```

The Sun Java System Application Server uses the same connection pool for both HADB session persistence and single sign-on. Single sign-on is enabled by default. If an application requires single sign-on functionality, the connection pool setting must be doubled.

TIP If your application does not require single sign-on functionality, disabling it can improve performance. To disable single sign-on, change the following settings in the `server.xml` file:

```
<virtual-server id="server1" ... >
    ...
    <property name="sso-enabled" value="false"/>
    ...
</virtual-server>
```

Solution

Double the size of the connection pool.

For example, if an application indicates that 16 is the optimal number of connections to a single HADB node, the number of connections should be doubled to 32 if single sign-on functionality is required. In this case, the JDBC connection pool settings look like this:

```
<jdbc-connection-pool steady-pool-size="32" max-pool-size="32"
max-wait-time-in-millis="60000" pool-resize-quantity="2"
idle-timeout-in-seconds="10000"
is-isolation-level-guaranteed="true"
is-connection-validation-required="true"
connection-validation-method="auto-commit"
fail-all-connections="false" datasource
classname="com.sun.hadb.jdbc.ds.HadbDataSource" name="CluJDBC"
transaction-isolation-level="repeatable-read">
```

You should also double the `loadbalancer.xml` file setting for `response-timeout-in-seconds` from 60 seconds to 12 0 seconds.

```
<property name="response-timeout-in-seconds" value="120"/>
```

This value must be equal to or greater than the following:

Max Response Time for any activity + (`<jdbc-connection-pool max-wait-time-in-millis="90000"`)

Server: Unable to obtain connection from pool

The application server is having trouble connecting with HADB, as evidenced by a message like the following in the server.log file:

```
ConnectionUtilgetConnectionsFromPool failed using connection URL:
null Unable to obtain connection from pool
```

Solution

Make sure HADB is running. Make sure that session-store, JDBC connection pool, and JNDI name (jdbc/hastore) are created. Configure the session persistence for High Availability with a command like the following:

```
asadmin configure-session-persistence --user admin
    --password netscape --host localhost --port 4848
    --type ha --frequency web-method --scope session
    --store jdbc/hastore server1
```

JDBC connection is not available

Consider the following:

- Is the max-pool-size setting adequate?

Is the max-pool-size setting adequate?

The `server.xml` file defines the following default values:

- `<ejb-container... : steady-pool-size="32"`
- `<jdbc-connection-pool...: max-pool-size="32"`

During server start/restart, the `ejb-container` steady pool for deployed enterprise beans will be created. Since the default `steady-pool-size` is 32, 32 enterprise beans will be created for each bean unless a different value is specified in the `sun-ejb-jar.xml` file.

The `setEntityContext` method will be called for each of the beans created. If more than one bean is grabbing JDBC connections in the `setEntityContext` method from the same JDBC connection pool, the following happens:

- During the steady pool creation, all the JDBC connections from the JDBC connection pool will be used (since `default max-pool-size` = 32)
- No connections will be left for any other beans that are created.

If the newly-created beans attempt to grab a JDBC connection from the same pool in their `setEntityContext` method, an exception is thrown with the following message:

```
No available resource . Wait-time expired.
```

Solution

Increase the `max-pool-size` of the default `jdbc-connection-pool` to a higher value, such as 256.

Exception occurs while calling DataSource.getConnection()

This exception occurs when an invalid DataSource class property is registered within the JDBC connection pool. Misspelling is a common cause. For instance, while creating a `jdbc-connection-pool` for Oracle, one might specify the following:

```
< property name="OracleURL" value="jdbc:oracle:...."/>
```

This will result in the following exception since `OracleURL` is not a property of any Oracle datasource:

```
NoSuchMethodException: setOracleURL
```

Solution

Verify that all `jdbc-connection-pool` properties in use are valid.

Verify that the datasource classname specified is for the required vendor datasource class.

Exception occurs while executing against MSSQL

The following exception occurs while executing a statement against the MSSQL server using a non-XA driver:

```
java.sql.SQLException: [DataDirect] [SQLServer JDBC Driver] Can't
start a cloned connection while in manual transaction mode
```

This happens when the `selectMethod` property is not set to `cursor`.

Solution

Ensure the `selectMethod` property is set correctly during JDBC connection pool registration:

```
<property name="selectMethod" value="cursor"/ >
```

Using the command-line interface, issue the following command:

```
asadmin set --user adminUser --password adminPassword
       --host hostID --port port#
       instanceName.jdbc-connection-pool.poolName
              .property.selectMethod="cursor"
```

The options can also be set using the graphical Administration interface.

IOException: Connection in invalid state

The error log shows the following message:

```
WEB2001: Web application service failed
java.io.IOException: Error from HA Store:
   Connection is in invalid state for LOB operations.
   <stack trace>
```

This error occurs when you have the HADB JDBC Connection pool `transaction-isolation-level` entry set to `read-committed` and `read-uncommitted`.

Solution

Change the `transaction-isolation-level` value of your HADB JDBC Connection pool to `repeatable-read` and re-start the application server.

Common Runtime Procedures

This section covers the following common runtime and recovery procedures:

- Improving CPU Utilization

Improving CPU Utilization

Available CPU cycles and I/O capacity can impose severe restrictions on performance. Resolving and preventing such issues is necessary to optimize system performance (in addition to configuring the HADB optimally.)

If there are more CPUs on the host that are not exploited, add new nodes to the same host. Otherwise add new machines and add new nodes on them.

If the machine has enough memory, increase the DataBufferPoolSize, and increase other internal buffers that may be putting warnings into the log files. Otherwise, add new machines and add new nodes on them.

For much more information on this subject, consult the *Performance and Tuning Guide*.

Chapter 5

HADB Issues on Windows

This section covers problems you may encounter when using Sun Java™ System Application Server 7 2004Q2 Update 1 Enterprise Edition with the High Availability Database (HADB) 4.4 on the Windows platform. HADB 4.4 has a new management architecture and new commands, compared to HADB 4.3 that is bundled with Application Server for UNIX platforms. For details on administering HADB 4.4, see *Sun Java System Application Server 7 2004Q2 Update 1 Administration Guide*.

Topics in this chapter include:

HADB Database Creation Fails

The error occurs when starting the database. The typical message in this case is:

```
failed to start database : HADB Database creation failed
```

To determine the cause of the problem, use the Log Viewer and/or inspect the *install_dir*/hadb/4/log directory. Some possible errors are:

No Available Memory

Description

Insufficient memory is available to create the database.

Solution 1

Check if there are other processes using up all memory the on Windows and end those processes if possible.

Solution 2

Install more memory in your system.

Review the documentation on shared memory requirements in the Preparing for HADB Setup chapter of the *Sun Java System Application Server Installation Guide.*

Too Few Semaphores

Description

HADB uses memory-mapped files for shared memory on Windows. You will get this message when there is not enough space on device disk for shared memory.

Solution

Make more space available on device disk for shared memory.

Database Nodes Cannot Be Reached and the Database Does Not Function

Solution

The IP addresses of the involved hosts should be static. If the addresses are dynamic (DHCP) the lease time should be set to forever (usually 0).

The Management Agents Could Not Establish a Domain

Description

The HADB management system is dependent on UDP Multicast messages on multicast address `228.8.8.8`. If these messages cannot get through, the `createdomain` command fails with the following message:

```
The management agents could not establish a domain, please check that the hosts can communicate with UDP multicast.
```

Possible causes include:

- The agents are running on hosts with several network interfaces on different subnets.
- There is a switch on the network that does not forward multicast messages.
- There is router on the network that does not route multicast messages with the address `228.8.8.8`.
- Multicast messages are disabled in the operating system.

Solution 1

If the hosts have several network interfaces on different subnets, the management agent must be configured to use one of the subnets. Set the `ma.server.mainternal.interfaces` attribute.

Solution 2

Configure the needed network infrastructure to support multicast messages.

Unexpected Node Restarts, Network Partitions, or Reconnects

Description

Unexpected node restarts, network partitions, or reconnects with messages "`Network Partition: *** Reconnect detected ***`" written in the HADB history files and on the HADB host terminals.

This may happen if multiple nodes identify themselves with the same physical node number.

Solution

Try stopping the database with the `hadbm stop` command, and look for "rogue" `hadb` processes on the hosts on which any HADB nodes have been running at any time. If there still are `hadb` processes running, these belong to rogue nodes.

On the hosts on which rogue nodes are found, check that the management agents are correctly configured, and that the management domain is correctly defined. There may be multiple management domains configured, and each host may possibly be included in more than one domain. Make sure that databases defined in separate domains do not have conflicting definitions, such as database nodes using the same port numbers.

`hadbm create` or `hadbm addnodes` Command Hangs

Description

Some hosts in the host list given to `hadbm create` or `addnodes` have multiple network interfaces, while others have only one, and the `hadbm create/addnodes` command hangs.

Solution

For the hosts having multiple network interfaces, specify the dotted IP address of the network interface (for example., `129.241.111.23`) to be used by `hadb` when issuing `hadbm create/addnodes`. If the host name is used instead of IP address, the first interface registered on the host will be used, and there is no guarantee that the nodes will be able to communicate.

ma (Management Agent Process) Crashes

Description

The ma (Management Agent process) crashes for various reasons.

Solution

Display diagnostic information by using hadbm listdomain.Typically, the remedy is to restart the failed agent. If that does not help, restart all agents in turn.

Server Responds Slowly After Being Idle

Description

The server takes a long time to service a request after a long period of idleness, and the sever log shows “lost connection” messages of the form:

```
java.io.IOException:..HA Store: Lost connection to the server.
```

In such cases, the server needs to recreate the JDBC pool for HADB.

Solution

Change the timeout value. The default HADB connection timeout value is 1800 seconds. If the application server does not send any request over a JDBC connection during this period, HADB closes the connection, and the application server needs to re-establish it. To change the timeout value, use the hadbm set SessionTimeout= command.

NOTE Make sure the HADB connection timeout is greater than the JDBC connection pool timeout. If the JDBC connection timeout is more than the HADB connection time out, the connection will be closed from the HADB side, but will remain in the appserver connection pool. When the application then tries to use the connection, the application server will have to recreate the connection, which incurs significant overhead.

Requests Are Not Succeeding

The following problems are addressed in this section:

Is the Load Balancer Timeout Correct?

Description

When configuring the `response-timeout-in-seconds` property in the `loadbalancer.xml` file, you must take into account the maximum timeouts for all the applications that are running. If the response timeout it is set to a very low value, numerous in-flight requests will fail because the load balancer will not wait long enough for the Application Server to respond to the request.

Conversely, setting the response timeout to an inordinately large value will result in requests being queued to an instance that has stopped responding, resulting in numerous failed requests.

Solution

Set the `response-timeout-in-seconds` value to the maximum response time of all the applications.

Are the System Clocks Synchronized?

Description

When a session is stored in HADB, it includes some time information, including the last time the session was accessed and the last time it was modified. If the clocks are not synchronized, then when an instance fails and another instance takes over (on another machine), that instance may think the session was expired when it was not, or worse yet, that the session was last accessed in the future!

NOTE In a non-co-located configuration, it is important to synchronize the clocks on that machines that are hosting HADB nodes. For more information, see the *Installation Guide* chapter, "Preparing for HADB Setup."

Solution

Verify that clocks are synchronized for all systems in the cluster.

Is the Application Server Communicating With HADB?

Description

HADB may be created and running, but if the persistence store has not yet been created, the Application Server will not be able to communicate with the HADB. This situation is accompanied by the following message:

```
WARNING (7715): ConnectionUtilgetConnectionsFromPool failed using
connection URL: connection URL
```

Solution

Create the session store in the HADB with a command like the following:

```
asadmin create-session-store --storeurl connection URL --storeuser haadmin
--storepassword hapasswd --dbsystempassword super123
```

Session Persistence Problems

The following problems are addressed in this section:

- The `create-session-store` Command Failed
- Configuring Instance-Level Session Persistence Did Not Work
- Session Data Seems To Be Corrupted

The `create-session-store` Command Failed

Description

The `asadmin create-session-store` command cannot run across firewalls. Therefore, for the `create-session-store` command to work, the application server instance and the HADB must be on the same side of a firewall.

The `create-session-store` command communicates with the HADB and not with the application server instance.

Solution

Locate the HADB and the application server instance on the same side of a firewall.

Configuring Instance-Level Session Persistence Did Not Work

The application-level session persistence configuration always takes precedence over instance-level session persistence configuration. Even if you change the instance-level session persistence configuration after an application has been deployed, the settings for the application still override the settings for the application server instance.

Session Data Seems To Be Corrupted

Description

Session data may be corrupted if the system log reports errors under the following circumstances:

- During session persistence
- When the session state is read during session activation
- When the session state is read after session failover

If the data has been corrupted, there are three possible solutions for bringing the session store back to a consistent state, as described below.

Solution 1

Use the `asadmin clear-session-store` command to clear the session store.

Solution 2

If clearing the session store does not work, re initialize the data space on all the nodes and clear the data in the HADB using the `hadbm clear` command.

Solution 3

If clearing the HADB does not work, delete and then recreate the database.

HADB Performance Problems

Performance is affected when the transactions to HADB get delayed or aborted. This situation is generally caused by a shortage of system resources. Any wait beyond five seconds causes the transactions to abort. Any node failures also cause the active transaction on that node at crash time to abort. Any double failures (failure of both mirrors) will make the HADB unavailable. The causes of the failures can generally be found in the HADB history files.

To isolate the problem, consider the following:

- Is There a Shortage of CPU or Memory Resources, or Too Much Swapping?
- Is There Disk Contention?
- Is There a Shortage of HADB Data Device Space?
- Is There a Shortage of Other HADB Resources?

Is There a Shortage of CPU or Memory Resources, or Too Much Swapping?

Description

Node restarts or double failures due to "`Process blocked for` *x* `sec, max block time is 2.500000 sec`." In this case, *x* is the length of time the process was blocked, and it was greater than 2.5 seconds.

The HADB Node Supervisor Process (`NSUP/clu_nsup_srv`) tracks the time elapsed since the last time it did some monitoring work. If that time duration exceeds a specified maximum (`2500ms` by default), NSUP concludes that it was blocked too long and restarts the node.

NSUP being blocked for more than 2.5 seconds cause the node to restart. If mirror nodes are placed on the same host, the likelihood of double failure is high. Simultaneous occurrence of the blocking on the mirror hosts may also lead to double failures.

The situation is especially likely to arise when there are other processes—for example, in a colocated configuration— in the system that compete for CPU, or memory which produces extensive swapping and multiple page faults as processes are rescheduled.

NSUP being blocked can also be caused by negative system clock adjustments.

Solution

Ensure that HADB nodes get enough system resources. Ensure also that the time synchronization daemon does not make large (not higher than 2 seconds) jumps.

Is There Disk Contention?

Description

A disk contention can have a negative impact on user data read / writes to the disk devices, as well as on HADB writing to history files. Severe disk contention may delay or abort user transactions. Delay in history file writing may cause node restarts and, in the worst case, lead to double failures.

The disk contention can be identified by monitoring the disk I/O from the OS, for the disks used for data, log devices and history files. This can also identified by the following statement in the history files: "`HADB warning: Schedule of async <read,write> operation took ...`"

History file write delays are written to the HADB history files. This can be identified by "`NSUP BEWARE` *timestamp* `Last flush took too long (`*x* `msecs)`."

This warning shows that disk I/O took too long. If the delay exceeds ten seconds, the node supervisor restarts the trans process with the error message:

```
Child process trans0 10938 does not respond.
Child died - restarting nsup.
Psup::stop: stopping all processes.
```

This message indicates that a `trans` (`clu_trans_srv`) process has been too busy doing other things (for example, waiting to write to the history file) to reply to the node supervisor's request checking the heartbeat of the `trans` process. This causes the `nsup` to believe that the `trans` has died and then restarts it.

This problem is observed especially in RH AS 2.1 when multiple HADB nodes are placed on the same host and all the nodes use the same disk to place their devices.

Solution

Use one disk per node to place the devices used by that node. If the node has more than one data devices and the disk contention is observed, move one data device to another disk. The same applies to the history file as well.

Is There a Shortage of HADB Data Device Space?

Description

One possible reason for transaction failure is running out of data device space. If this situation occurs, HADB will write warnings to the history file, and abort the transaction which tried to insert and/or update data.

Typical messages are:

```
HIGH LOAD: about to run out of device space, ...
HIGH LOAD: about to run out of device space on mirror node, ...
```

The general rule of thumb is that the data devices must have room for at least four times the volume of the user data. Please refer to the *Tuning Guide* for additional explanation.

Solution 1

Increase the size of the data devices using the following command:

```
hadbm set TotalDataDevicePerNode=size
```

This solution requires that there is space available on the physical disks which are used for the HADB data devices on all nodes.

HADBM automatically restarts each node of the database.

Solution 2

Stop and clear the HADB, and create a new instance with more nodes and/or larger data devices and/or several data devices per node. Unfortunately, using this solution will erase all persistent data. See the *Administrator's Guide* for more information about this procedure.

See Bug ID 5097447 in the "Known Problems" section of the *Application Server 7 Release Notes* for more information.

Is There a Shortage of Other HADB Resources?

When an HADB node is started, it will allocate:

- Several shared memory segments of fixed size
- Internal data structures of fixed size

If an HADB node runs out of resources it will delay and/or abort transactions. Resource usage information is shipped between mirror nodes, so that a node can delay or abort an operation which is likely to fail on its mirror node.

Transactions that are delayed repeatedly may time out and return an error message to the client. If they do not time out, the situation will be visible to the client only as decreased performance during the periods in which the system is short on resources.

These problems frequently occur in "High Load" situations. For details, see "High Load Problems."

High Load Problems

High load scenarios are recognizable by the following symptoms:

- User requests do not succeed
- The database gives multiple timeout and “transaction aborted” messages
- Frequent “`HIGH LOAD`” warnings in the history file
- Sporadic failures

If a high load problem is suspected, consider the following:

- Is the Tuple Log Out Of Space?
- Is the `node-internal` Log Full?
- Are There Enough Locks?
- Can You Fix the Problem by Doing Some Performance Tuning?

NOTE Frequently, all of these problems can be solved by making more CPU horsepower available.

Is the Tuple Log Out Of Space?

All user operations (`delete`, `insert`, `update`) are logged in the tuple log and executed. There tuple log may fill up because:

- Execution slows due to CPU or disk I/O contention
- The mirror node is slow in receiving the log records (“`log throw due to...`” messages in the history files), which can happen as a result of:
  - Network contention, so the log records do not reach the mirror node
  - CPU contention at the mirror node, which keeps it from processing the received log records quickly enough.

Solution 1

Check CPU usage, as described in “Improving CPU Utilization”.

Solution 2

If CPU utilization is not a problem, check the disk I/O. If the disk shows contention, avoid page faults when log records are being processed by increasing the data buffer size with `hadbm set DataBufferPoolSize=...`

Solution 3

Look for evidence of network contention, and resolve bottlenecks.

Solution 4

Increase the tuple log buffer using `hadbm set LogBufferSize=...`

See Bug ID 5097447 in the "Known Problems" section of the *Application Server 7 Release Notes* for more information.

Is the `node-internal` Log Full?

Too many node-internal operations are scheduled but not processed due to CPU or disk I/O problems.

Solution 1

Check CPU usage, as described in "Solution 2: Improve CPU Utilization."

Solution 2

If CPU utilization is not a problem, and there is sufficient memory, increase he `InternalLogbufferSize` using the `hadbm set InternalLogbufferSize=` command.

Are There Enough Locks?

Some extra symptoms that identify this condition are:

- Error code `2080` or `2096` delivered to the client.
- `HIGH LOAD` log messages thrown due to locks.
- `hadbm resourceinfo --locks` shows locks allocated, and all are in use., all the time

Solution 1: Increase the number of locks

Use `hadbm set NumberOfLocks=` to increase the number of locks.

Solution 2: Improve CPU Utilization

Check CPU usage, as described in "Improving CPU Utilization".

Can You Fix the Problem by Doing Some Performance Tuning?

In most situations, reducing load or increasing the availability of resources will improve host performance. Some of the more common steps to take are:

- Run the nodes on hosts with better hardware characteristics (more internal memory, higher processor speed, more processors).
- Add physical disks and use several data devices, not more than one device on each physical disk.
- Add more nodes, on new hosts, and refragment the data to utilize the new nodes.
- Change configuration variables to allocate larger memory segments or internal data structures.

In addition, the following resources can be adjusted to improve "`HIGH LOAD`" problems, as described in the *Performance and Tuning Guide*:

Table 5-1 HADB Performance Tuning Properties

Resource	Property
Size of Database Buffer	`hadbm attribute DataBufferPoolSize`
Size of Tuple Log Buffer	`hadbm attribute LogBufferSize`
Size of Node Internal Log Buffer	`hadbm attribute InternalLogBufferSize`
Number of Database Locks	`hadbm attribute NumberOfLocks`

Client cannot connect to HADB

Description

This problem is accompanied by a message in the history file:

```
HADB-E-11626: Error in IPC operations, iostat = 28: No space left on device
```

where:

- 11626 is an HADB error code, "Error in IPC operations," which means that some Inter Process Communication operation failed.
- "iostat = 28" means (on Solaris) that the operating system set errno to ENOSPC, which again translates to "No space left on device."

If HADB started successfully, and you get this message at runtime, it means that the host computer has too few semaphore undo structures.

Solution

Stop the affected HADB node, reconfigure and reboot the affected host, restart the HADB node. HADB will be available during the process.

Improving CPU Utilization

Description

Available CPU cycles and I/O capacity can impose severe restrictions on performance. Resolving and preventing such issues is necessary to optimize system performance (in addition to configuring the HADB optimally.)

Solutions

If there are additional CPUs on the host that are not exploited, add new nodes to the same host. Otherwise add new machines and add new nodes on them.

If the machine has enough memory, increase the DataBufferPoolSize, and increase other internal buffers that may be putting warnings into the log files. Otherwise, add new machines and add new nodes on them.

For more information on this subject, consult the *Performance and Tuning Guide*.

HADB Administration Problems

The hadbm command and its many subcommands and options are provided for administering the high-availability database (HADB). The hadbm command is located in the *install_dir*/SUNWhadb/4/bin directory.

Refer to the chapter on Configuring the High Availability Database in the *Sun Java System Application Server Administrator's Guide* for a full explanation of this command. Specifics on the various `hadbm` subcommands are explained in the `hadbm` man pages.

The following problems are addressed in this section:

`hadbm` Command Fails: `The agents could not be reached`

Description

The command fails with the error:

`The agents` <*url*> `could not be reached.`

The hosts in the URL could be unreachable either because the hosts are down, because the communication pathway has not been established, because the port number in the URL is wrong, or because the management agents are down.

Solution

Verify that the URL is correct. If the URL is correct, verify that the hosts are up and running and are ready to accept communications; for example:

```
ping hostname1
ping hostname2
...
```

hadbm Command Fails: command not found

Description

The hadbm command can be run from the current directory, or you can set the search PATH to access the hadb commands from anywhere, which is much more convenient. The error, "hadbm: Command not found," indicates that neither of these conditions has been met.

Solution 1

cd to the directory that contains the hadbm command and run it from there:

```
cd install_dir/SUNWhadb/4/bin/
./hadbm
```

Solution 2

Use the full path to invoke the hadbm command:

```
install_dir/SUNWhadb/4/bin/hadbm
```

Solution 3

You can use the hadbm command from anywhere by setting the PATH variable. Instructions for setting the PATH variable are contained in the "Preparing for HADB Setup" chapter of the *Sun Java System Application Server 7 Installation Guide.*

To verify that the PATH settings are correct, run the following commands:

```
which asadmin
which hadbm
```

These commands should echo the paths to the utilities.

hadbm Command Fails: JAVA_HOME not defined

Description

The message "Error: JAVA_HOME is not defined correctly" indicates that the JAVA_HOME environment variable has not been set properly.

Solution

If multiple Java versions are installed on the system, ensure that the `JAVA_HOME` environment variable points to the correct Java version (1.4.1_03 or above for Enterprise Edition).

Instructions for setting the `PATH` variable are contained in the "Preparing for HADB Setup" chapter of the *Sun Java System Application Server 7 Installation Guide.*

`create` Fails: "`path does not exist on a host`"

Description

After issuing the `hadbm create` command, an error similar to the following appears on the console:

```
./hadbm create ...
...
hadbm:Error 22022: Specified path does not exist on a host. Please specify
a valid path: [ machineName ... ]
```

This error message indicates that the HADB server component is not installed on the machine on which you are trying to create the HA database.

Solution

Log in to the host and create paths for the HADB devices and HADB history files. Run `hadbm create` and specify the `--devicepath` and `--historypath` options to the paths created. Also make sure that the user running the management agent on the host has read and write access to these directories.

NOTE HADB executables cannot be installed on different paths on different hosts.

Database Does Not Start

The `create` or start command fails with the console error message:

```
hadbm: Error 22095: Database could not be started...
```

Consider the following possibilities:

- Was there a shared memory get segment failure?

Was there a shared memory get segment failure?

Description

The history files show the error message:

```
..'systemerr'..HADB-S-01760: Shared memory get segment failed..
```

Solution 1

Reboot the system.

Solution 2

If the problem persists, the operating system may not have enough shared memory or semaphores. Increase them according to the number of nodes in the machine. (For details, see the *Deployment Guide*). Note that after making these changes, the machine must be restarted to make them available.

Do the History Files Contain Errors?

Description

If the problem still persists, inspect the HADB history files. Some of the more likely error messages to look for are:

- `Shared memory get segment failed`

  The system has not been set up with enough shared memory. (Discussed in the previous section.)

- `Could not verify node address`

  This message occurs when another process is using the port that an HADB server is trying to process. It can occur in several situations:

  - The `portBase` is used by another process running on this host machine.

    Set the `PortBase` attribute to another value using the following command:

    `hadbm set portbase=`*`value`*

  - An attempt to stop the HADB node for maintenance failed.

Try again to stop the node with the `hadbm` command. If that fails, use Windows `Task Manager` to end the OS process, `clu_nsup_srv`, for this node. The `nsup` process should then end all its HADB child processes. If the `nsup` process does not exist you have to remove all the HADB child processes one by one.

- The HADB node was stopped for maintenance and an `inetd` process restarted the HADB node before you intended to start it.

  Make sure that `inetd` does not start the HADB node before stopping it.

- `hadbm` *command* `fails with internal error: "The database could not be started"`

  Check the following:

  - Shared memory is correct on all machines in the HADB configuration.
  - No other HADB databases are running on the machines, or any other processes that could be using the same port numbers.
  - All necessary directories exist and have write permissions.
  - There is enough space in directory where devices are going to be written.

Solutions

After verifying that none of the above errors have occurred, try the following remedies, in order:

- Delete the database and retry.
- Delete the database, reboot, and retry.
- Delete database, reinstall the HADB software, and retry.
- Contact Product Support.

For more information, refer to the *Error Message Reference*.

Do You Need a Simple Solution?

As a last resort, try the following possible solutions.

Solution 1

Delete the database with the `hadbm delete` command, and see if that allows the `hadbm create` to proceed normally.

Solution 2

Sometimes a system reboot is the necessary last resort. Issue `hadbm delete`, reboot the machine, and then rerun the `hadb create` command.

`clear` Command Failed

When this command fails, the history files are likely to explain why. See "Do the History Files Contain Errors?" on page 118 for instructions on viewing the history files and a list of some common error messages.

`create-session-store` Failed

The `asadmin create-session-store` command could fail for one of these reasons:

- `Invalid user name or password`
- `SQLException: No suitable driver`

Invalid user name or password

This error occurs when the `--dbsystempassword` supplied to the `create-session-store` command is not the same password as the one given at the time of database creation.

Solution 1

Try the command again with the correct password.

Solution 2

If you cannot remember the `dbsystem` password, you need to clear the database using `hadbm clear` and provide a new `dbsystem` system password.

SQLException: No suitable driver

The `create-session-store` produces the error: `SessionStoreException: java.sql.SQLException: No suitable driver`.

Solution 1

This error can occur when `asadmin` is not able to find `hadbjdbc4.jar` from the `AS_HADB` path defined in `asenv.conf` in the Application Server `config` directory.

The solution is to change `AS_HADB` to point to the location of the HADB installation.

Here is a sample `AS_HADB` entry from an `asenv.conf` file:

```
AS_HADB=c:\install_dir\SUNWhadb\4.4.0-8
```

Solution 2

This error can also occur if you provide the incorrect value for `--storeUrl`. To solve this problem, obtain the correct URL using `hadbm get jdbcURL`.

Attaching Shared Memory Segment Fails Due To Insufficient Space

Description

The server throws an error message like the following:

```
Attaching shared memory segment with key xx failed,
OS status=12 OS message: Not enough space.
```

Solution

Increase shared memory.

Cannot Restart the HADB

Description

HADB restart does not work after a double node failure. Additional recovery actions are needed before HADB can be restarted.

Symptoms of a double node failure include:

- `hadbm` status shows that the HADB status is non-functional.
- The node status shows that the nodes are in `Starting` or `Recovering` state. Even after stopping and then restarting each of the nodes, they remain in the `Starting` state. Eventually, the node status changes to `Stopped`.

This problem occurs when mirror HADB host machines have failed or been rebooted, typically after a power outage, or when a machine is rebooted without first stopping the HADB (in a single-machine installation), or when a pair of mirror machines from both Data Redundancy Units (DRUs) are rebooted.

If mirror host machine pairs are rebooted, or if host failures cause an unplanned reboot of one or more mirror host machine pairs, then the mirror nodes on these machines are not available, and the data is likely to be in an inconsistent state, because a record may have been in the process of being committed when the power failed, or the reboot occurred.

TIP To prevent such problems, be sure to use the procedure described in the HADB chapter of the *Administration Guide* when rebooting as a part of a planned maintenance.

HADB cannot heal itself automatically in such "double failure" situations because the part of the data that resided on the pair nodes is lost. In such cases, the `hadbm start` command does not succeed, and the `hadbm status` command shows that HADB is in a non-operational state.

Explanation

For performance reasons, the HADB does much of its data management in memory. If both DRUs are rebooted, then the HADB does not have a chance to write its data blocks to disk.

For more information on the DRUs and HADB confutation, see "Administering the High Availability Database" in the *Administration Guide*, and the *Deployment Guide*.

TIP If the HADB exhibits strange behavior (for example consistent timeout problems), and you want to check whether a restart cures the problem, use the `hadbm restart` command.

When the HADB is restarted in this manner, HADB services remain available. Conversely, if HADB is started and stopped in separate operations using `hadbm stop` and `hadbm start`, HADB services are unavailable while HADB is stopped.

Solution

1. Follow the instructions under "Recovering from Session Data Corruption" in the "Administering the High Availability Database" chapter of the *Administration Guide*.
2. Verify that the node states show `Starting`/`Recovering`, then reset the database.

Chapter 6

Administration Problems

This chapter discusses problems that you may encounter while administering the Application Server. Full reference material and instructions for performing administration tasks can be found in the *Administrator's Guide* and *Administrator's Guide to Security*.

The following sections are contained in this chapter:

Server Logs

This section covers:

Application Server logs

The Application Server collects and stores event information in two log files which are located in the logs directory:

- `access`—Application Server instance HTTP events
- `server.log`—Application Server events

Log entries can also be directed to another log file as specified by the administrator. In addition, each virtual server within an Application Server instance has its own identity and can have its own log file.

The following components and subsystems can utilize selective logging of server messages:

- CORBA-based clients (ORB)
- Web container
- Enterprise JavaBeans (EJB) container
- Message-driven bean (MDB) container
- Java Transaction Service (JTS)
- Java Message Service (JMS)
- Virtual Servers

Extensive information on how these logs work and the information gathered in them is available in the Using Logging chapter of the *Sun Java System Application Server Administrator's Guide*. Log levels are also described in the online help of the Administration interface.

HADB History Files

Inspection of the history files is a common procedure described under "Examining the HADB history files" on page 149.

General Problems

This section covers the following problems:

- All administrative operations fail

All administrative operations fail

A message like the following appears while performing adminstrative operations:

```
Manual changes to [server1] configuration detected.
Please reconfigure (Keep or Discard manual changes).
```

All subsequent operations then fail, except for the operation that gets you out of this situation.

This problem arises when users edit any of the major configuration files manually, without using the administrative interfaces (Admin Console or CLI).

Explanation

In reality, there are two sets of configuration files, one used by the server, and one that the administrative interfaces manipulate. The existence of two files requires the system to "apply" any changes you make using an administrative interface. For example, if you create a resource with the `asadmin create-resource` command, the server does not see that resource until you reconfigure the server.

The administrative server continuously examines the following configuration files of a server instance `<install_dir>/config`:

- `server.xml` - Configuration for all the J2EE subsystems and resources, applications etc. One per instance.
- `init.conf` - Configuration for core Web Engine. One per instance.
- `<virtual-server-id>.obj.conf` - Virtual Server configuration. One per virtual server. Hence could be multiple of these per instance.
- `mime.types` - Configuration for all the mime types that the server supports. Could be multiple per instance.

If any of these files is manually modified, the administrative server blocks users from doing any administrative tasks, to prevent any possibility of conflict between the manual modification and changes made through administrative interface.

Important Note:
The administration server uses the files in the `<instance_dir>/config/backup` directory, rather than the configuration files that the server runtime uses, in `<instance_dir>/config`. Never modify the backup files by hand. If you happen to change the actual configuration files llisted above, you will have to use the `reconfigure` command in asadmin.

Solution

To resolve this situation following actions can be taken. They are presented in order of safety, so each one requires an increased degree of caution.

1. Be absolutely sure that the manual modification is correct from both XML validation point of view (it should resolve correctly against the DTD) and general verification point of view (for example, you have not set an undeployed web module as the default web module of a virtual server).
2. Start `asadmin` in multi or single mode. In single mode you generally need to specify administrative server credentials on the command line .
3. Issue one of the following `reconfigure` commands:
   - Issue `reconfigure --keepmanualchanges` *`server-id`*
     where *`server-id`* is the target server whose configuration was modified. This command will change the administration server's view of the server's configuration and modify itself to use the manual changes. Use this option only if you are absolutely sure that your manual changes were correct. Both the servers will then be in sync.
   - Or issue `reconfigure --discardmanualchanges` *`server-id`*
     where *`server-id`* is the target server whose configuration was modified. This command changes the target server's view of its own configuration. In most cases, the administration server will notify the target server (if running) of the changes, and the changes will be dynamically applied. The administration server and target server will then be in sync.
4. If you are absolutely sure about the changes that you made to the real configuration files, then you can take following actions—but they are *not* recommended. They are to be used only as a last resort, and only then by a power user:
   - Stop both the administrative server and target server.
   - Overwrite the modified files (for example, `server.xml` and `init.conf`) in `<`*`instance_dir`*`>/config/backup directory` with the files in `<`*`instance_dir`*`>/config directory`.
   - Delete `<`*`instance_dir`*`>/config/backup/server.xml.timestamp` and `<`*`instance_dir`*`>/config/backup/init.conf.timestamp` .

Command-Line Interface Problems

This section discusses problems that you may encounter while using the command-line interface of the Application Server.

Can't access the command-line utility.

After installing the Application Server software, you will need to configure your environment to include the `bin` directory of the Application Server if you are going to do any of the following:

- Run the `asadmin` command
- Run the command-line utility
- Access the `asant` utility for working with the sample applications.

Solution

Add the *install_dir*`/bin` directory to your PATH environment variable. If you are not familiar with the process of setting environment variables, refer to the post-installation instructions in the *Sun Java System Application Server Installation Guide*.

NOTE If your Admin Server is running under SSL, the `--secure` flag must be used.

Can't access the Application Server man pages.

For the Solaris unbundled version of the product, you will not be able to access the man pages until you add the *install_dir*`/man` to the MANPATH environment variable.

Solution

Add *install_dir*`/man` to your MANPATH environment variable.

Graphical Interface Problems

This section discusses problems that you may encounter while using the Administration interface of the Application Server.

NOTE If your Admin Server is running under SSL, https: / /... must be used for browser access.

This section addresses the following issues:

- Can’t access the Administration interface.
- Can’t undo accidental “changes.”

Can’t access the Administration interface.

If the connection was refused when attempting to invoke the graphical Administration interface, it is likely that the Admin Server is not running.

Solution

Refer to “Can't access the Admin Server.” on page 57 for information on troubleshooting this problem.

Can’t undo accidental “changes.”

If an instance has been flagged for Apply Changes Required, and you decide NOT to make changes (perhaps the changes were a mistake and you want to forget the whole thing), there is no obvious method to *unset* the Apply Changes Required condition. Clicking Apply Changes seems to be forced at this point.

Shutting down your browser, restarting the Application Server instance, and so on. does NOT clear the Apply Changes flag. You are still prompted to apply the changes (since the backup configuration file is different from the current and applied configuration file).

Monitoring Problems

This section covers:

- Load Balancer Plug-in isn't being monitored

Load Balancer Plug-in isn't being monitored

Logging for the load balancer plug-in is not automatically turned on. To turn on load balancer plug-in log messages:

1. Set the web server logging level to DEBUG.
2. Set the value of the `require-monitor-data` property to true. For example:

```
<property name="require-monitor-data" value="true" />
```

TIP When logging is enabled on the load balancer plug-in, the load balancer writes HTTP session IDs in the web server log files. Therefore, if the web server hosting the load balancer plug-in is located in the DMZ, we recommend that you do not use the DEBUG or similar log level in production environments. If you must use the DEBUG logging level, then you should turn off load balancer logging by setting the `require-monitor-data` property to false in `loadbalancer.xml` file.

For more information, refer to the Configuring Load Balancer chapter of the Sun Java System Application Server *Administrator's Guide*.

Authentication/Authorization Problems

This section addressed the following problems:

- Don't know the admin username / password
- Don't know the Admin Server port number
- Connection Refused when accessing the Admin Server
- Can't import the certificate for my server.
- The server does not recognize my certificate.
- LDAP authentication / equalization is not working.

Don’t know the admin username/password

You don’t have the admin username or password you need to administer the system.

Solution 1

Try the user name `admin`. This is the default user name specified in the server configuration dialog during installation. Typical passwords are `adminadmin`, or `administrator`.

Solution 2

Examine the following file (assuming your admin server is under domain1):

Solaris: `/sun/appserver7/domains/domain1/admin-server/config/admpw`

Windows 2000:
`D:\Sun\AppServer7\domains\domain1\admin-server\config\admpw`

The file consists of a single line such as:

```
admin:{SHA}W6ph5Mm5Pz8GgiULbPgzG37mj9g=
```

The first field (before the colon) is the user name and the second field is the encrypted password. Although you can’t read the password, you can see the username, which may jog your memory.

Note:
The `config` directory that contains the `admpw` file can be accessed only by the user who installed the product.

Solution 3

Change the username and reset the password to nothing by modifying the `admpw` file (assuming your admin server is under domain1):

To change the username, type it in place of the existing name. To reset the password, delete all text after the colon. Then save the file and restart the admin server. You can now log in to the admin UI using the specified username, with no password. You should then immediately set a new password by navigating to Admin Server -> Security -> Access Control.

Solution 4

Delete the administrative domain and recreate it with a new password.

Solution 5

As a last resort, uninstall and reinstall the Application Server.

Don’t know the Admin Server port number

If you do not know the HTTP server port number of the Admin Server, you can inspect the Admin Server's configuration file to determine the HTTP server port number:

1. Navigate to *domain_config_dir*`/domain1/admin-server/config/` and open the `server.xml` file in a text editor.
2. Look for the following element:

   ```
   http-listener id="http-listener-1" address="0.0.0.0" port="4848"...
   ```

   Here, port 4848 is the HTTP port number used by the admin server.

Connection Refused when accessing the Admin Server

If the connection is refused when attempting to access the Admin Server with your browser, it is likely that the Admin Server is not running.

Solution

Start the Admin Server, if you have not already done so, using the instructions in “Starting and Stopping the Server” in the *Installation Guide*. Otherwise consult the Admin Server log file to determine why it failed to start, as described in “Using Logging” in the *Administrator’s Guide*..

Can’t import the certificate for my server.

Consider the following:

- Has the trust database been created?
- Was the certificate generated with the right tool?

Has the trust database been created?

If you haven't created the trust database in Sun Java System Application Server, you need to do that.

Solution

In the Security page of the Administration interface, click the Manage Database tab and create the trust database by entering its password.

Was the certificate generated with the right tool?

The app server supports NLS database only. So, `certutil` and `openssl` are compatible tools. You can't use certificates generated by `keytool` directly on Appserver.

Solution

Generate the certificate with `certutil` or `openssl`.

The server does not recognize my certificate.

There are three certificates involved in client certificate authentication.

1. First is the server certificate with which you will enable security in the server instance. This must be installed in the server as a Certificate for "This server."
2. Second is the client certificate which you will install in the browser to authenticate yourself to the server when client-cert authentication is enabled.
3. Third is the server certificate chain which links the prior two certificates. This must be installed in the server instance as the certificate for "Server certificate chain." If this certificate is not installed on the server instance, the instance doesn't know which client certificate to authenticate.

Solution

Verify that all the certificates have been implemented correctly. Be sure that you implement the chain in #3 and that the ROOT Certificate Authority (CA) is trusted.

LDAP authentication/equalization is not working.

In order for the Application Server to use an LDAP-based directory server for authentication and authorization, the security realm must be configured and the LDAP realm must be activated.

Solution

1. In the left pane of the Administration interface, expand the server1 / Security / Realms / ldap tree.

2. In the right panel, verify that the Classname field contains the following information:

   `com.iplanet.ias.security.auth.realm.ldap.LDAPRealm`

   This class is the interface between the Application Server and the LDAP-based directory server.

3. Click Properties to display the pane for configuring specifics for the Directory Server implementation. Enter data similar to the following:

   - Name: directory Value: ldap://localhost:389
   - base-dn Value: dc=sun,dc=com
   - jaas-context Value ldapRealm

4. In the left pane of the Administration interface, expand the server1/Security hierarchy and change the Default Realm to ldap.

5. Apply Changes and Restart your instance as prompted.

HADB Administration Problems

In the Sun Java System Application Server 7, Enterprise Edition, the `hadbm` and its many subcommands and options is provided for administering the high-availability database (HADB). A summary of the `hadbm` commands in contained in "Summary of High Availability Commands" on page 175.

The `hadbm` command is located in the *install_dir*`/SUNWhadb/4/bin directory`.

Refer to the chapter on Configuring the High Availability Database in the Sun Java System Application Server *Administrator's Guide* for a full explanation of this command. Specifics on the various `hadbm` subcommands are explained in the `hadbm` man pages.

The following problems are addressed in this section:

hadbm command fails: host unreachable.

The command fails with the error, “Host unreachable: <hostname>”.

The host could be unreachable either because it is down, or because the communication pathway has not been established. To isolate the problem, consider the following:

- Is the host up and running?
- Is RSH or SSH set up and running?
- Are the SSH binaries in the proper location?
- Is your communication protocol configured properly?

Is the host up and running?

If the remote host isn’t running or can’t accept connections, attempts to access it will fail.

Solution

Try pinging the host to see if it is up and running, ready to accept communications:

```
ping <hostname>
```

Is RSH or SSH set up and running?

The communication pathway must be established before the `hadbm` command can succeed.

Solution

The `hadbm` commands will not work if host communication has not been set up. That is, the HADB nodes must have been configured for Remote Shell (RSH) or Secured Shell (SSH). Refer to “Preparing for HADB setup” in the Installation Guide for guidelines on verifying RSH and SSH.

If the verification does not work, remote communication for the cluster has not been set up correctly. Instructions for doing this are contained in the Setting Up Host Communication section of the *Sun Java System Application Server Installation Guide.*

Are the SSH binaries in the proper location?

When using SSH, the relevant binaries must be in the proper location.

Solution

Make sure that the `ssh` and `scp` binaries are in `/usr/bin`.

Is your communication protocol configured properly?

Your communication protocol (RSH / SSH) must be configured properly.

Solution

If you are using `clsetup`, and you plan to use RSH for your communication, make sure you uncomment the following line in the `clresource.conf` file:

```
set  managementProtocol=rsh
```

If you are using SSH, make sure you closely follow all the SSH configuration steps contained in the *Sun Java System Application Server Installation Guide.*

hadbm command fails: command not found

The `hadbm` command can be run from the current directory or you can set the search `PATH` to access the `hadb` commands from anywhere, which is much more convenient. The error, “hadbm: Command not found”, indicates that neither of these conditions has been met.

Solution 1

You can `cd` to the directory that contains the `hadbm` command and run it from there:

```
cd <install_dir>/SUNWhadb/4/bin/
hadbm
```

Solution 2

You use the full path to invoke the `hadbm` command:

```
<install_dir>/SUNWhadb/4/bin/hadbm
```

Solution 3

You can use the `hadbm` command from anywhere by setting the PATH variable. Instructions for setting the PATH variable are contained in the Preparing for HADB Setup chapter of the Sun Java System Application Server *Installation Guide*.

To verify that the PATH settings are correct, run the following commands:

```
which asadmin
which hadbm
```

These commands should echo the paths to the utilities.

hadbm command fails: JAVA_HOME not defined

The message “Error: JAVA_HOME is not defined correctly” indicates that the JAVA_HOME environment variable has not been set properly.

If multiple Java versions are installed on the system, you must ensure that the JAVA_HOME environment variable points to the correct Java version (1.4.1_03 or above for Enterprise Edition).

Instructions for setting the PATH variable are contained in the Preparing for HADB Setup chapter of the Sun Java System Application Server *Installation Guide*.

create fails: “path does not exist on a host”

After issuing the `hadbm create` command, an error like the following appears on the console:

```
./hadbm create ...
...
hadbm:Error 22022: Specified path does not exist on a host. Please
specify a valid path: [ machineName ... ]
```

This error message indicates that the HADB server component is not installed on the machine on which you are trying to create the HA database.

Solution

Install the HADB server in the in the <*install_dir*> directory, and run the command again.

NOTE HADB executables cannot be installed on different paths on different hosts.

---

database doesn't start.

The `create` or start command fails with the console error message:

```
hadbm: Error 22095: Database could not be started...
```

Consider the following possibilities:

- Was there a shared memory get segment failure?
- Do the history files contain errors?
- Do you need a simple solution?

Was there a shared memory get segment failure?

If the history files show the error message:

```
..'systemerr'..HADB-S-01760: Shared memory get segment failed..
```

Solution 1: Use sync;sync and reboot instead of init 6

The `hadbm create` command can fail with this error occurs after making changes to `/etc/system` and doing a system reset with the `init 6` command.

Instead of re-spinning the machine with `init 6`, do `sync;sync` as `root` user and then `reboot`.

Solution 2: Increase the amount of shared memory

There may not be as much shared memory as the HADB needs. The amount of shared memory required by HADB depends on parameters like `DataBufferPoolSize`, `LogbufferSize`, and other parameters. Look into the file `/etc/system` and set `shmsys:shminfo_shmmax` to the maximum value possible (the preferred value is `0xffffffff`).

Verify that other shared memory settings are configured correctly. After making your changes, issue the `hadbm stop` command and (for Solaris) reboot the machine. (For Linux, rebooting is not necessary.)

For more information on the mechanics of configuring shared memory, consult the chapter, "Preparing for HADB Setup" in the *Sun Java System Application Server Installation Guide*. For guidelines on choosing the best settings, consult the *Performance Tuning Guide*.

Solution 3: Verify /etc/system settings

Verify the settings in the system file. Even a single mistyped character can create problems.

Solution 4: Resolve conflicts

Use `ipcs` to see if there are any shared memory segments or semaphores occupied unnecessarily by you or the other users. Use `ipcrm` to free them and then try starting the database.

Solution 5: Increase the number of semaphores

If the problem persists, then the operating system may not have enough shared memory or semaphores, etc. Increase them according to the number of nodes you have in the machine. (For details, see the Deployment Guide). Note that after making these changes, you must restart the machine to make them available.

Do the history files contain errors?

If the problem still persists, inspect the history files, as described in "Examining the HADB history files" on page 149.

Some of the more likely error messages to look for are:

- `Shared memory get segment failed`
  The system has not been set up with enough shared memory. (Discussed in the previous section.)
- `Could not verify node address`
  This message occurs when another process is using the port that an HADB server is trying to process. It can occur in several situations:
  - The portBase is used by another process running on this host machine. In that case, set the PortBase attribute to another value using the command `hadbm set portbase=<value>`.
  - You tried to stop the hadb node for maintenence, but that action failed. Try again to stop the node with the `hadbm` command. If that fails, `kill` the OS process `clu_nsup_srv` for this node *without* the `-9` option. The `nsup` process should then stop its hadb child process. If the parent process nsup does not exist, kill all the child processes using `kill -9`. (For more information, see "Is There a Disk Contention?" on page 86.)
  - You stopped the hadb node for maintenence and an `inetd` process restarted the hadb node before you intended to start it.
    In that case, make sure that `inetd` does not start the hadb node before you stop it by following the instructions in "Stopping a node when inetd is active" on page 153.
- `hadbm <command>' fails with internal error:`
  `"The database could not be started"`

  Check the following:

- RSH and SSH are set up correctly, and you can communicate with all the machines in the HADB configuration.
- Shared memory is all correct on all machines in the HADB configuration.
- No other HADB databases are running on the machines, or any other processes that could be using the same port numbers.
- All necessary directories exist and have write permissions.
- There is enough space in directory where devices are going to be written.

Once you've verified that none of the above errors have occurred, try the following remedies, in order:

- Delete the database and retry.
- Delete the database, reboot, and retry.
- Delete database, reinstall the HADB software, and retry.
- Contact Product Support, as described on page 21.

For more information, refer to the Error Message Reference.

Do you need a simple solution?

As a last resort, try the following possible solutions.

Solution 1: Delete the database

Issue the `hadbm delete` command, and see if that allows the `hadbm create` to proceed normally.

Solution 2: Reboot the machine.

Sometimes a system reboot is the necessary last resort. Issue `hadbm delete`, reboot, and then rerun the `hadb create` command.

clear command failed

When this command fails, the history files are likely to explain why. See "Do the history files contain errors?" on page 140 for instructions on viewing the history files and a list of some common error messages.

create-session-store failed

The `asadmin create-session-store` command could fail for one of these reasons:

- `Invalid user name or password`
- `SQLException: No suitable driver`

Invalid user name or password

This error occurs when the `--dbsystempassword` you supplied to `create-session-store` command is not the same password as the one given at the time of database creation.

Solution 1

Try the command again with the correct password.

Solution 2

If you can't remember the dbsystem password, you'll need to clear the database using `hadbm clear` and provide a new dbsystem system password.

SQLException: No suitable driver

The `create-session-store` produced the error: `SessionStoreException: java.sql.SQLException: No suitable driver`.

Solution 1

This error can occur when asadmin is not able to find `hadbjdbc4.jar` from the `AS_HADB` path defined in the `asenv.conf` in application server `config` directory.

To solve it, change the `AS_HADB` to point to the location of your HADB installation

Here is a sample `AS_HADB` entry from an `asenv.conf` file:

```
AS_HADB=/export/home0/hercules/0815/SUNWhadb/4.2.2-17
```

Solution 2

This error can also occur if you provide the incorrect value for `--storeUrl`. To solve that problem, obtain the correct URL using `hadbm get jdbcURL`.

Attaching shared memory segment fails due to insufficient space

You get an error message like the following:

```
Attaching shared memory segment with key xx failed,
OS status=12 OS message: Not enough space.
```

Solution: Increase shared memory

See "Solution 2: Increase the amount of shared memory" on page 139.

Can't restart the HADB

HADB restart will not work after a double node failure. Additional recovery actions are needed before HADB can be restarted.

Symptoms of a double node failure include:

- hadbm status shows that the HADB status is non-functional.
- The node status shows shows that the nodes are in Starting or Recovering state. Even after stopping and then restarting each of the nodes, they remain in the Starting state. Eventually, the node status changes to Stopped.

This problem occurs when mirror HADB host machines have failed or been rebooted, typically after a power outage, or when a machine is rebooted without first stopping the HADB (in a single-machine installation), or when a pair of mirror machines from both Data Redundancy Units (DRUs) are rebooted.

If mirror host machine pairs are rebooted, or if host failures cause an unplanned reboot of one or more mirror host machine pairs, then the mirror nodes on these machines are not available—and the data is likely to be in an inconsistent state, because a record may have been in the process of being committed when the power failed, or the reboot occured.

TIP To prevent such problems, be sure to use the procedure described in the HADB chapter of the Admin Guide when rebooting as a part of a planned maintenance.

HADB cannot heal itself automatically in such 'double failure' situations, because the part of the data that resided on the pair nodes is lost. In such cases, `hadbm start` command does not succeed, and the `hadbm status` command shows that HADB is in a non-operational state.

Explanation

The HADB does much of its data management in memory, for performance. If both DRUs are rebooted, then the HADB doesn't have a chance to write its data blocks to disk.

For more information on the DRUs and HADB configuation, see "Administering the High Availability Database" in the Administration Guide, and the Deployment Guide.

TIP If you notice strange HADB behavior (for example consistent timeout problems) and want to check whether a restart cures the problem, use the `hadbm restart` command.

When you restart the HADB in that manner, data remains available. On the other hand, if you stop and start HADB in separate operations using `hadbm stop` and `hadbm start`, data is unavailable while HADB is stopped.

Solution

1. In the Admin Guide, in the chapter "Administering the High Availability Database", follow the instructions under, "Recovering from Session Data Corruption".
2. If other parts of the system are running, take the steps described in "Maintaining service while taking the HADB offline" on page 150.
3. Verify that the node states show Starting/Recovering, then reset the database by following the instructions in "Clearing the database and recreating the session store" on page 151.

Error: Specified database does not exist

This error message occurs when the management host you are using to issue the hadbm command is different from the management host that was used to create the HADB .

Solution

Use the same host that was used when the HADB is created.

hadbm command doesn't return control to user.

Many hadbm commands, in particular hadbm set, restart all the nodes of the database in order. If some problem has occurred, then the command may not return.

Solution 1

From another window / shell, look at the history files for all the nodes to see if an error has occurred or if the command is still in progress. Run `hadbm status --nodes` to see if all the nodes are up and running. If they are not and there appears to be a permanent failure, you will need to cancel the command, and then try running `hadbm restart`.

Solution 2

If Solution 1 fails, and your command was an attempt to set a configuration value for hadbm, try resetting it back to its old value and see if the database restarts correctly.

If the restart continues to fail, follow the instructions in "Clearing the database and recreating the session store" on page 151 to reset the database.

Solution 3

If clearing the database is unsuccessful, you'll have to delete the database using `hadbm delete`, recreate it using `hadbm create`, and then recreate the session store using `asadmin create-session-store`.

Cluster Administration Problems

In the Sun Java System Application Server 7, Enterprise Edition, you can use the `cladmin` command to run the following `asadmin` commands simultaneously on all application server instances in a cluster: `start-instance`, `stop-instance`, `deploy`, `undeploy`, `create-jdbc-resource`, `create-jdbc-connection-pool`, `configure-session-persistence`, `delete-jdbc-resource`, `delete-jdbc-connection-pool`. This simplifies the task of cluster administration.

The `cladmin` command is located in the *install_dir*`/bin` directory. The default location of the `cladmin` input files, `clinstance.conf` and `clpassword.conf`, is `/etc/opt/SUNWappserver7`.

Refer to the chapter on Using the cladmin Command in the Sun Java System Application Server *Administrator's Guide* for a full explanation of this command.

This section addresses the following problems:

- Refragmentation of the HADB fails.
- The cladmin command is not working.
- Application is not available on the cluster.

Refragmentation of the HADB fails.

The attempt to refragment the HADB failed.

Consider the following possibility:

- Is there enough space on the data devices?

Is there enough space on the data devices?

Messages like these indicate that refragmentation failed for lack of space on the data devices:

```
HIGH LOAD: about to run out of device space ...
HIGH LOAD: about to run out of device space on mirror node ...
```

The problem occurs when data devices are filled beyond 50% or 60% of the available space, which does not leave enough extra space to carry out the refragmentation.

TIP Monitor your data device usage using the `hadbm deviceinfo` command.

Solution 1: Make more unreserved blocks available on data devices

Use the `df` command to see how much space has been used on the machine. To determine the space available for user data, take 99% of the total device size, then subtract 4 times the LogBufferSize. The difference between the total device size and the free size is the user data size. If the data may be refragmented in the future, the user data size should not exceed 50% of the space available for user data. If refragmentation is not relevant, close to 100% may be used.

Solution 2

Find out what disk the data devices are on with the `hadbm get DevicePath` command and check the for space on that disk. If there is room, increase the size of the data devices using the following command:

```
hadbm set TotalDataDevicePerNode=size
```

If the data devices cannot accommodate a copy of the user data during refragmentation, then refragmentationwill not succeed. If the refragmenting is performed while adding nodes, you will need to delete the database and create a new database including the new nodes. In that case, the data is lost.

The cladmin command is not working.

Consider the following possibilities:

- Are the Admin Servers of all the instances in the cluster started?
- Do all the instances in the cluster have same administrator user name and password?
- Are the input files correct?
- Are the input files on all instances in the cluster identical?

Are the Admin Servers of all the instances in the cluster started?

Before running the `clsetup` command, all the Admin Servers in the cluster must be running.

Do all the instances in the cluster have same administrator user name and password?

During installation, the installation program creates a `clinstance.conf` file with entries for two instances. If you add more instances to the cluster, you must add information about these instances in the `clinstance.conf` file.

Are the input files correct?

The order in which entries appear in the `clinstance.conf` file is important and must not be changed from the default order. If you add information about more application server instances, entries for these instances must in the correct order.

Solution

Verify that any changes you have made to the input files follow the format specified in the *Sun Java System Application Server Administrator's Guide.*

Are the input files on all instances in the cluster identical?

The values in the input files must be identical on all instances in the cluster. The `cladmin` command is not designed to set up each instance with different values.

Solution

Verify that the `cladmin` input files are identical on all instances in the cluster.

Application is not available on the cluster.

Consider the following possibilities:

- Did the application deploy successfully to the cluster?

Did the application deploy successfully to the cluster?

It's possible that the deploy operation failed. To find out, run this command against each instance in the cluster.

```
asadmin list-components --type web
```

Solution

If the application isn't listed, try redeploying it and look for errors during deployment.

Common Administration and Recovery Actions

This section describes common administrative and recovery procedures that are used in a variety of situations.

This section covers:

- Examining the HADB history files

- Maintaining service while taking the HADB offline
- Starting the hadb nodes after rebooting the host machine
- Clearing the database and recreating the session store
- Stopping a node when inetd is active
- Rebooting a machine that has HADB nodes

Examining the HADB history files

The history files are generally found at their default location, `/var/tmp`. If they are not at that location, use `hadbm get HistoryPath` to find the path to the history files.

The history file names are of the form `<dbname>.out.<nodenumber>`. The default database name is `hadb`, so for the default database name, the history file for node 0 would be `hadb.out.0`.

For example, for an HADB instance named failover, with two nodes on the same system, the history file names would be `failover.out.0` and `failover.out.1`.

NOTE The `hadbm` utility cleans up all the files it created when `hadbm create` fails. In that case, the messages about the cause of the error are lost. But if the client machine has a `historypath` directory (default `/var/tmp`), then the history files are preserved there when `the command` fails.

If the historypath directory does not exist, you need to examine the syslogs on the hosts for error and warning messages from HADB. Messages are prefixed with "HADB" (the default syslogprefix value, which can be changed using the `create` command's `--set` option).

For additional information on the history files, refer to the Configuring the High Availability Database chapter in the *Administrator's Guide.*

Maintaining service while taking the HADB offline

Any command that makes HADB unavailable (such as hadbm clear) causes the application servers to start reporting errors in the error log. Client requests will then take a long time to get handled as the application continues retrying its requests to HADB, which can't answer because it is unavailable.

You can avoid this situation by disabling session persistence prior to clearing or stopping the database. This procedure takes time, but it lets the system maintain full service of your application(s) while HADB is down.

Perform the following steps:

1. Disable session persistence by using `cladmin` to set `availability-enabled` to `false` for the cluster. (See the "Session Persistence" section of the Admin Guide for the details of this procedure.)
2. Restart all your instances using the following approach:
   - Disable half of the instances in your cluster (or as many as you can at a time to maintain the necessary level of service for your application) by marking them as disabled in the load balancer configuration file. (See the load balancing section of the Admin Guide for details
   - After the quiescence period has been reached, restart the disabled instances, and then re-enable them in the load balancer.
   - Repeat those steps for the next batch of instances until you have restarted all the instances
3. Once HADB is back up and running again, set availability-enabled to true and follow the restart process again.

Starting the hadb nodes after rebooting the host machine

When one or more HADB host machines undergoes an unplanned reboot, use the `hadbm status` command to see if the machines hosted HADB mirror nodes. If they did, then:

- If the database status is “functional”, then the hadb nodes on the rebooted machines have to be restarted after the host machines have come up. Use `hadbm status –nodes` to find out which nodes are not running and which machines host those nodes, then start the nodes using `hadbm startnode` on their host machines.
- If the database status is “non-functional”, see “Can’t restart the HADB” on page 143.

TIP You may want to restart a node if you notice strange behavior in a node (for example excessive CPU consumption) and want to check whether a restart cures the problem. Use the `hadbm restartnode` command to restart an HADB node.

Clearing the database and recreating the session store

Clearing the database, restarting it, and recreating the session store is always the quickest way to fix your database. All session data in HADB will be lost, but all session data will still be available because it exists in the application server cache. (The only exception is sessions that have been passivated. They will not be in the application server cache and thus will be lost when you clear HADB.)

TIP If you need to keep servicing user requests, following the instructions in the previous section, “Maintaining service while taking the HADB offline”.

TIP **Avoid losing important data.**

For transient session data, losing the data is generally not an issue. The problem concerns losing passivated session state, that is, when a session no longer has any data in the Application Server cache and the state has been passivated to the HADB. Sessions are passivated when the maximum number of sessions exceeds the number specified in `sun-web.xml` file for each application.

To avoid losing data, configure the application to reject any new session requests when the maximum number of sessions is met by setting the maximum number of sessions to a very high value and rejecting any sessions beyond that. This prevents passivation, thus avoiding the risk of losing session data if you need to clear the HADB.

1. Use this command to clear the database, reinitialize all data devices, and recreate all system tables:

   `hadbm clear --spares` *`x`* `--dbpassword=`*`tttt`* `smokedb`

   where *x* is the number of spares you originally had, and *tttt* is the database password.

   This command clears the database—all your old data is lost.

2. Get the JDBC URL:

   ```
   asadmin hadbm get jdbcURL smokedb
   ```

3. Recreate the appserver schemas and set up session persistence:

   ```
   asadmin create-session-store
   --storeurl <jdbc url returned from step 4>
   --storeuser appservusr --storepassword <password>
   ```

   When issuing this command, make sure the user and password used in create-session-store match the user and password specified in the JDBC connection pool for HADB.

   For more information, see the *Administrator's Guide.*

Stopping a node when inetd is active

If you have set up `inetd`, a node is automatically restarted when you issue the `stopnode` command The command `hadbm status --nodes` then shows the node as starting, even though you just stopped it. The node then resumes the running state, but is in the offline role.

To make changes to the host which require a reboot, you need to perform some additional tasks to stop the node:

1. Comment out the `inetd` entry for that node from the `inetd` configuration files (otherwise, the node is automatically restarted as soon as you stop it).
2. Re-add the entry to the `inetd` files after you have restarted the node.
3. Restart `inetd` by sending a `SIGHUP` to the process. For example:

```
ps -e |grep inetd (to find PID)
kill -HUP <PID_inetd>
```

For additional information, refer to the HADB Setup chapter of the *Administrator's Guide*.

Rebooting a machine that has HADB nodes

HADB achieves fault tolerance by replicating data on mirror nodes. Mirror nodes should be placed on separate Data Redundancy Units in a deployment environment. (See the Admin Guide, chapter "Administering the High Availability Database" and the Deployment Guide for details on DRUs and HADB configurations.)

NOTE Single machine configurations are recommended *only* for development and test environments.

HADB tolerates single point failures—the failure of one node, the failure of one machine, the simultaneous failure of multiple machines belonging to the same DRU, or the failure of the whole DRU. However, HADB does not tolerate double failures—simultaneous failure of one or more mirror machine pairs.

For that reason, you need to exercies care when rebooting a machine that has HADB nodes. For a complete description of the procedure to follow , see the *Administration Guide*, HADB chapter, "Maintaining the HADB machines".

Chapter 7

Application and Deployment Problems

This chapter addresses problems that you may encounter while deploying applications in the Application Server environment.

The following sections are contained in this chapter:

Application Problems

This section covers:

persistence.support.JDODataStoreException

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

EJB Problems

This section covers the following issues:

Can’t set TRANSACTION_SERIALIZABLE level for a bean

Since the Release Notes say that `check-modified-at-commit` flag is not implemented for this release, is there an equivalent of the Weblogic `<transaction-isolation>` element in Sun Java System Application Server?

Solution

First identify which resource is being used in that bean, then add the following attribute to the `jdbc-connection-pool` in the `server.xml` file:

```
transaction-isolation-level="serializable"
```

This is an optional element and does not exist in the `server.xml` file by default. You must explicitly add it.

You can do this either using the Administration interface or using the command-line interface to modify the `server.xml` file, then running the `asadmin reconfig` command.

Runtime exception from ejbCreate()

An "Unable to create reference" runtime exception is generated by `ejbCreate` when it tries to create a instance for a composite primay key.

Explanation

Every entity bean must have unique primary key. The primary key class must obey the following rules:

- The primary key class must be Serializable.
  The client and server can then exchange keys using by-value semantics.
- The primary key class must override the `equals` and `hashCode` methods.
  The server uses these methods to see if a bean is already resident in memory, so proper implementation is crucial! If you get it wrong or forget to override the `equals` or `hashCode` methods(violation of the EJB 2.0 specification) the bean fails at runtime for a variety of seemingly bizarre reasons.

Solution

The Entity bean's primary key class must provide suitable implementation of the `hashcode()` and `equals(Object other)` methods to simplify the management of the primary keys by the client code. (The implementation of these methods is also mandated by the EJB 2.0 specification, in section 12.2.12.)

CMP Problems

This section covers the following issues:

- Are you using a supported JDBC driver?

Are you using a supported JDBC driver?

For a list of the supported JDBC drivers for the Application Server, refer to the *Sun Java System Application Server Platform Summary*.

Parser exceptions generated by verifier

This section discusses the SAX Parser exceptions generated by the verifier and how to interpret them. It covers the following exceptions:

- Element type "sun-ejb-jar" is not declared.
- Root element type is "sun-ejb-jar", but was declared to be "ejb-jar"
- java.net.UnknownHostException: www.sun.com
- Expected "</name>" to terminate element starting on line 8.
- Element "ejb" does not allow "jndi-name" here.

NOTE The parser exceptions generated for the `sun-web.xml` file have a different error statement owing to the fact that the Schema2Beans framework is used to create the XML graph. The error messages described here follow the "Cause By: " line when verifying that file.

Element type "sun-ejb-jar" is not declared.

The error message looks like this:

```
Error: ** Error trying to process file:
com.iplanet.ias.deployment.AppConfigException: Error parsing
sun-specific ejb deployment descriptor, line 5, Element type
"sun-ejb-jar" is not declared.

com.iplanet.ias.deployment.AppConfigException: Error parsing
sun-specific ejb deployment descriptor, line 5, Element type
"sun-ejb-jar" is not declared.

        at
com.iplanet.ias.deployment.EjbBundleXmlReader.load(EjbBundleXmlRead
er.java:265)

        at
com.iplanet.ias.deployment.EjbBundleXmlReader.loadStdAloneModule(Ej
bBundleXmlReader.java:162)

        ....
```

There are two probable causes of this exception. The exception usually points to the root element. The most notable cause is the fact that the DOCTYPE has not been declared. Hence in case of a validating parser, there is no DTD to validate against and the error message is spurted out. Another common problem is the PUBLIC id that is being specified. The validating parser looks for the DTD by resolving the path that is mapped to the string specified in the PUBLIC id. Hence if the string points to a different DTD you will get such an exception because the specified root elements will differ. Another inference of the error is the fact that the root tag for the XML does not match the one that is specified in the DTD.

Root element type is "sun-ejb-jar", but was declared to be "ejb-jar"

The error message looks like this:

```
Error: ** Error trying to process file:
com.iplanet.ias.deployment.AppConfigException: Error parsing
sun-specific ejb deployment descriptor, line 6, Root element type is
"sun-ejb-jar", but was declared to be "ejb-jar".

com.iplanet.ias.deployment.AppConfigException: Error parsing
sun-specific ejb deployment descriptor, line 6, Root element type is
"sun-ejb-jar", but was declared to be "ejb-jar".

        at
com.iplanet.ias.deployment.EjbBundleXmlReader.load(EjbBundleXmlRead
er.java:265)

        at
com.iplanet.ias.deployment.EjbBundleXmlReader.loadStdAloneModule(Ej
bBundleXmlReader.java:162)

        ...
```

The most probable cause of this parser exception is the fact that the root element declared in the DOCTYPE for the XML does not match the root element in the DTD furnished. The root element is the starting tag of the xml. This is compared with the root element that is declared in the DOCTYPE. The DTD is got from the PUBLIC id which is resolved to point to the physical DTD.

java.net.UnknownHostException: www.sun.com

The error message looks like this:

```
Error: ** Error trying to process file:
com.iplanet.ias.deployment.AppConfigException: Error parsing
sun-specific ejb deployment descriptor, line 4, External entity not
found:
"http://www.sun.com/software/sunone/appserver/dtds/sun-ejb-jar_2_0-
0.dtd".

com.iplanet.ias.deployment.AppConfigException: Error parsing
sun-specific ejb deployment descriptor, line 4, External entity not
found:
"http://www.sun.com/software/sunone/appserver/dtds/sun-ejb-jar_2_0-
0.dtd".
```

```
        at
com.iplanet.ias.deployment.EjbBundleXmlReader.load(EjbBundleXmlRead
er.java:265)

        at
com.iplanet.ias.deployment.EjbBundleXmlReader.loadStdAloneModule(Ej
bBundleXmlReader.java:162)

        ......
```

Or it may look like this:

```
Error: ** Error trying to process file:
com.iplanet.ias.deployment.AppConfigException: Error creating
sun-specific web xml graph

com.iplanet.ias.deployment.AppConfigException: Error creating
sun-specific web xml graph

        at
com.iplanet.ias.deployment.WebBundleXmlReader.createIasWebDescripto
rObj(WebBundleXmlReader.java:297)

        at
com.iplanet.ias.deployment.WebBundleXmlReader.load(WebBundleXmlRead
er.java:257)

        ......
```

The error commonly occurs if the Entity pointed to by the URL does not exist.

Another common problem that cuases the exception is the absence of the internet connection to connect to the URL The Application Server has a catalog mechanism whereby all the DTDs required for validating parsing are mapped to their physical locations present in the `appserv-rt.jar`. So you should never get this exception if you are validating a standard J2EE or a Sun specific deployment descriptor.

If you do get the exception, then the cause is a mismatch in the PUBLIC id that is specified in the file, and the one in the DTD. The `String.equals()` match is done for the mapping resolution, so the PUBLIC id must exactly match the PUBLIC id for the intended DTD. Even a tiny error like leaving out a comma can cause this exception.

Expected "</name>" to terminate element starting on line 8.

The error message looks like this:

```
Error: ** Error trying to process file:
com.iplanet.ias.deployment.AppConfigException: Error parsing
sun-specific ejb deployment descriptor, line 8, Expected "</name>"
to terminate element starting on line 8.

com.iplanet.ias.deployment.AppConfigException: Error parsing
sun-specific ejb deployment descriptor, line 8, Expected "</name>"
to terminate element starting on line 8.

        at
com.iplanet.ias.deployment.EjbBundleXmlReader.load(EjbBundleXmlRead
er.java:265)

        at
com.iplanet.ias.deployment.EjbBundleXmlReader.loadStdAloneModule(Ej
bBundleXmlReader.java:162)

        .....
```

This exception shows that line 8 is missing terminator name tag, which violates XML's well-formedness constraints. Since XML is case sensitive., it is also possible to have a starting name tag and ending name tag that have the same letters, but which do no match in case as, for example, `<name>. . . </NAME>`. The exception can also be caused by not closing an empty tag with a "/" at the end, as in `<img .. >` instead of the correct `<img ... />`.

Element "ejb" does not allow "jndi-name" here.

The error message looks like this:

```
Error: ** Error trying to process file:
com.iplanet.ias.deployment.AppConfigException: Error parsing
sun-specific ejb deployment descriptor, line 10, Element "ejb" does
not allow "jndi-name" here.

com.iplanet.ias.deployment.AppConfigException: Error parsing
sun-specific ejb deployment descriptor, line 10, Element "ejb" does
not allow "jndi-name" here.

        at
com.iplanet.ias.deployment.EjbBundleXmlReader.load(EjbBundleXmlRead
er.java:265)

        at
com.iplanet.ias.deployment.EjbBundleXmlReader.loadStdAloneModule(Ej
bBundleXmlReader.java:162)

        .......
```

There are two reasons for such an exception. The first reason is that the tag specified in the error statement, here jndi-name is not declared in the DTD or the tag specified has the wrong spelling or the case.

Another common and reason is the sequence of the tag specified in the DTD. If the DTD specifies that the tag `ejb` is defined as

```
<!ELEMENT ejb (ejb-name, jndi-name?,...
```

then we cannot define `jndi-name` before the `ejb-name` tag. (Note: DTD validation checks case sensitivity as well as the sequence of the tags defined in the DTD.)

Ant Problems

This section covers the following issues:

- Can’t run Ant remotely.

Can’t run Ant remotely.

To run Ant from a remote machine, the Ant utility must be installed on that machine. You can deploy to the remote machine using the Administration interface, and you can use the `asadmin` command or the Sun ONE Studio for deployment on the remote machine.

Additionally, the default settings that are set up during Application Server installation assume the local instance to be the target for all administrator operations from `asant` (deployment, resource registration, and so on). This can be changed by editing the *install_dir*`/samples/common.properties` file. This file is referred to by all the samples that use `common.xml`, that is, most of the samples. A variety of settings can be customized: admin host, port, user, password, target instance, and so on.

Solution

Edit the file to set the values for the remote machine or instance and run the following command on the sample.

```
asant deploy
```

Chapter 8

Integration Problems

This chapter discusses problems that may occur between the Application Server product and an associated product or component.

The following sections are contained in this chapter:

Web Server Problems

This section addresses the following types of web server problems:

Sun ONE Web Server Problems

For Sun ONE Web Server, the load balancer DTD and XML files should be located in the *webserver_install_dir*/*webserver_instance*/`config` directory.

Apache Web Server Problems

For Apache Web Server, the load balancer DTD and XML files should be located in the *webserver_install_dir*/*webserver_instance*/`conf` directory.

Reverse Proxy Plug-In Problems

Check to be sure that the file path for `libpassthrough.so/passthrough.dll` specified in the `magnus.conf` file is correct.

Database Problems

This section addresses the following problems:

- Pooled connection to Sybase
- Oracle XA Exception: Resource Manager doing work outside a global transaction

Pooled connection to Sybase

Attempting to configure the Application Server for pooled connection to Sybase., produces errors like the following:

```
CORE3283: stderr: at....createDataSource(Unknown Source) ....
CORE3283: stderr: at...createResource(Unknown Source)
....
```

The errors occur for both `com.sybase.jdbc2.jdbc.SybDataSource` and `com.sybase.jdbc2.jdbc.SybConnectionPoolDataSource`.

Solution

Sybase properties in `server.xml` must be configured exactly as expected, no more properties, no less properties: user, password, portNumber, serverName and databaseName.

Here is an example of a correct configuration:

```
<jdbc-connection-pool steady-pool-size="8" max-pool-size="32"
   max-wait-time-in-millis="60000"
   pool-resize-quantity="2"
   idle-timeout-in-seconds="300"
   is-isolation-level-guaranteed="true"
   is-connection-validation-required="false"
   connection-validation-method="auto-commit"
   fail-all-connections="false"
   datasource-classname="com.sybase.jdbc2.jdbc.SybDataSource"
```

```
   name="sbConnectionPool">
<property name="user" value="your_sybase_user"/>
<property name="password" value="your_sybase_user_password"/>
<property name="portNumber" value="your_sybase_port_number"/>
  <property name="serverName" value="your_sybase_server_name"/>
  <property name="databaseName" value="your_sybase_database_name"/>
</jdbc-connection-pool>
```

Oracle XA Exception: Resource Manager doing work outside a global transaction

When the resource manager is doing work outside a global transaction, you see an exception like this:

WARNING: JTS5041:
The resource manager is doing work outside a global transaction javax.transaction.xa.XAException ...

Solution 1

Check your oracle version, if it is 8.1.6, there are separate patches that need to be installed. Check the Oracle website for details. (Oracle 8.1.7 and above is recommended for XA support.)

Solution 2

In `server.xml`, make sure the `jdbc-connection-pool` element has the correct property settings to access the database. Make sure that the database is configured correctly, as well.

Messaging Problems

This section addresses the following types of messaging problems:

- JMS broker won't start.
- SOAP FAULT with Apache SOAP 2.3.1 client

JMS broker won’t start.

Refer to “Restart operation fails” on page 65 for information on this problem.

SOAP FAULT with Apache SOAP 2.3.1 client

When using Apache SOAP 2.3.1 client for accessing web-services installed on the application server, you must set the target namespace for the service to work (even though the client can access the services installed on other servers).

The error occurs because, whenever the Application Server is invoked, it checks for the namespace of the service. If the namespace is empty it returns a SOAP FAULT.

Solution

While a request like the following may work on other servers :

```
<?xml.version='1.0'.encoding='UTF-8'?>
<SOAP-ENV:Envelope.xmlns:SOAP-ENV=
   "http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<SOAP-ENV:Body >
   < ns1:sayHello.xmlns:ns1=" ";
   SOAP-ENV:encodingStyle=
      "http://schemas.xmlsoap.org/soap/encoding/" >
   <String_1.xsi:type="xsd:string"> Duke! </String_1>
```

The form required by this Application Server is:

```
<?xml.version='1.0'.encoding='UTF-8'?>
<SOAP-ENV:Envelope.xmlns:SOAP-ENV=
   "http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<SOAP-ENV:Body>
   <ans1:sayHello xmlns:ans1="http://hello.org/wsdl">
   <String_1.xsi:type="xsd:string"> Duke! </String_1>
```

Chapter 9

Migration Problems

This chapter discusses problems you may encounter while migrating applications to the Application Server environment.

The following sections are contained in this chapter:

- Migration Toolbox doesn't run correctly
- Non-fatal error occurs during extraction
- Fatal error occurs during extraction
- Translation error occurs
- Trouble migrating from WebLogic

Full instructions for performing migration tasks can be found in the *Sun Java System Application Server Migrating and Redeploying Server Applications* manual.

Migration Toolbox doesn't run correctly

If you have difficulty running the Sun Java System Application Server Migration Toolbox, consider the following possibilities:

- Is the script customized for your environment?
- Are JAR files in your extension directory?
- Are you using the correct JDK?

Is the script customized for your environment?

It is important that the `%MIGTBX_HOME%/bin/setenv.bat` script be customized for your environment. Because of limitations of the JDK, you cannot install the Migration Toolbox in a path containing directory names with spaces. For example, do not unpack the archive in your `C:\Program Files` directory. Instead, unpack the archive either in `c:\AppServer7` or `c:\`.

Are JAR files in your extension directory?

To avoid class version issues, all JAR files should be removed from your JDK extension directory (`%JAVA_HOME%/jre/lib/ext`) while running the Migration Toolbox. All the classes necessary for running the Migration Toolbox are included with the distribution.

NOTE Simply renaming the JAR files in the extension directory is not sufficient; you must move them to a different location.

Are you using the correct JDK?

Because many development machines have several installed copies or versions of the JDK, be sure you know which copy of the JDK you are using. Set the JAVA_HOME environment variable in the `%MIGTBX_HOME%/bin/setenv.bat` file to ensure you are running the preferred copy with the Migration Toolbox application.

Non-fatal error occurs during extraction

If only part of the automated migration succeeds (or fails), implement one of the solutions in "Migration Toolbox doesn't run correctly" on page 167, then rerun the extraction or translation. If that doesn't work, consider the following possibilities:

- Are you doing a NetDynamics migration?
- Is manual migration warranted?

Are you doing a NetDynamics migration?

If a problem occurs with a NetDynamics migration, create a new project in the NetDynamics Studio and import the problematic objects. Simplify them until you can get this project to run through the appropriate tools. Introduce these files back into the original, now-migrated, project.

Is manual migration warranted?

You may need to migrate the failed objects by hand. This is not as difficult as it may sound. The JATO framework was also designed for manual application authoring. Using the templates in the application package, follow the example of a migrated object of the same type. Documentation has been created to assist in creating new JATO objects manually.

Fatal error occurs during extraction

If a fatal error occurs, verify that the following items are not factors in the failure (listed in approximate order of likelihood):

1. Incorrect environment settings—Check the settings of your `%MIGTBX_HOME%/bin/setenv.bat` file and ensure they are appropriate for your machine.
2. Missing external classes
3. Incorrect tool property settings—Ensure that the Extraction Tool has valid property settings.
4. Using a non-existent runtime feature in a critical location, such as a class initializer or initialization of non-Spider threads to perform background tasks
5. Missing links directory or corrupted class files
6. Using the incorrect JDK version or platform
7. Conflicting class file versions in the boot classpath, such as those present in the JDK's extension directory

Translation error occurs

If you encounter an error during application translation, do the following:

- Verify that your application description file looks complete and that the XML is valid. Use a tool like `XMLSpy` or Internet Explorer to open the document and view it.
- Verify that the Translation Tool settings are correct.
- Verify that your environment settings in the `%MIGTBX_HOME%/bin/setenv.bat` file are appropriate for your machine.
- Verify that you have a complete Migration Toolbox installation.

Trouble migrating from WebLogic

User is trying to port a WebLogic 6.1 web application to Sun Java System Application Server 7 using the Application Server Migration Toolbox. The job is nicely done.

But now two things occur when using a web application:

- it don't know if the webapp is connecting with the Oracle DB
- in debug mode the AppServ gave me this:

LONG message...

Possible Solution

1. Copy the `struts.jar` file into `/WEB-INF/lib` and make the TLD and DTD files valuables on this directory.
2. Rebuild the EAR file with the `*.sh`.

It seems that the Migration Toolbox cannot handle JSP files that contain struts and other custom tags. These files do not get copied to the target area by the Migration Toolbox, and are listed as errors. I'm assuming that these files can just be moved manually, and the server then configured with the appropriate JAR files (`struts.jar`, and so on). I was just wondering if anyone had already got past this tool limitation, and if they'd be willing to share their experiences on migrating struts components into Sun Java System Application Server 7.

Chapter 10

Upgrade Problems

This chapter addresses problems that you may encounter while upgrading from one version of the Application Server to another.

The following sections are contained in this chapter:

Full instructions for upgrading can be found in the upgrading chapter of the *Sun Java System Application Server 7, Standard Edition Installation Guide.*

Upgrade Log

Upgrade events and errors are captured in the Application Server upgrade log file (`upgrade.log`) during the active upgrade process. For commonly-encountered errors, the most likely action needed is included in this file for your convenience.

Package-based installation consistently fails.

Examine the package installation logs to determine what is happening.

User ID errors occur.

Errors specific to the user ID (for tarball installations) usually have to do with permissions not being set correctly.

Solution

Verify that your permissions for the installation directory and the backup directory are correct.

Errors occur during the backup.

The most likely error in this phase is related to permissions and space limitations.

Solution

- Verify that your permissions for the directory specified are correct.
- Resolve any space problems, then restart the upgrade program where it left off.

Errors occur during reconfiguration.

Problems in this phase are usually caused by a system crash or other interruption.

Solution

After resolving any problems, restart the upgrade program where it left off.

Upgrade interrupted; server is in inconsistent state.

You can restart an upgrade process for any reason by removing the `.audit_upgrade` file in the directory where the `setup` script is located, then restarting the upgrade. However, until you have completed the upgrade, the server instance you are upgrading is in an inconsistent state, so it is important that you complete any upgrade you have started.

Solution

Complete the upgrade as quickly as possible to resolve the inconsistent state of the server.

Doing a partial online upgrade of a cluster causes problems.

You must upgrade all application server instances in a cluster together. Otherwise there is a risk of version mismatch caused by a session failing over from one instance to another where the instances have different versions of components running.

NOTE You cannot perform an online upgrade of a cluster if the upgrade involves a change to the application database schema or the HADB schema. This requires a full reinstall.

Solution

Upgrade all instances in a cluster identically and in as short a time period as possible.

Additional information on upgrading a cluster online is contained in the *Sun Java System Application Server Administrator's Guide.*

Doing a partial online upgrade of a cluster causes problems.

Appendix A

Summary of High Availability Commands

As a convenience, this appendix contains a summary of the command-line interface high-availability database (HADB) management commands for the Sun Java System Application Server 7, Enterprise Edition product.

- For detailed syntax on the `hadbm` commands, refer to the man pages.
- For instructions and guidelines on using the `hadbm` commands, refer to the *Sun Java System Application Server Administrator's Guide*. Instructions for using the `clsetup` command are contained in the *Installation Guide*.
- For a complete listing of the `asadmin` commands, refer to the appendix on Using the Command Line Interface in the *Administrator's Guide*.
- The `hadbm` command is located in the *install_dir*`/SUNWhadb/4/bin` directory.

The `hadbm` commands are run from the command-line using the following format:

```
hadbm <subcommand>
```

Table A-1 shows the available subcommands.

Table A-1 High-Availability Commands

`*clear`	Reinitializes all the data space on all nodes, and clears all user data.
`list`	Lists the existing HADBs.
`nodestatus`	Provides the key status information on all the nodes for the named HADB.
`refragment`	Stores data on a newly-created node.
`resourceinfo`	Provides information about the HADB resources.
`restart`	Restarts the HADB
`restartnode`	Restarts the node in the specific start level.

`set`	Sets the value of the specified configuration attributes to the named values.
`start`	Starts the HADB after it has been stopped.
`startnode`	Starts the node by running the startup procedure on the node.
`status`	Provides information on the state of the HADB.
`stop`	Stops the HADB
`stopnode`	Stops the node if the mirror node is up.

Frequently Asked Questions (FAQs)

These frequently-asked-questions (FAQs) in this appendix are sorted into the following sections:

Installation and Access

This section addresses the following questions:

How to tell which version of the application server is running?

Execute the following command:

```
asadmin version
```

How to determine the build date and version?

To get the application server version, you use this command:

```
asadmin version
```

To get the build date, use this variation:

```
asadmin version --verbose
```

The build date appears in parentheses, like this:

```
... (build A021926-226218)
```

The field before the hyphen contains the date, and the field after the hyphen contains the time. The following examples show ho to extract the date and time from those fields.

To extract the date from “A021926”:

- The first 2 digits in the field (02) are the last two digits of the year (2002).
- Subtract 11 from the next two digist (19) to get the month (8, or August).
- Subract 11 from the last two digits (26) to get the day (15).

Here, the build date is 08/15/2002.

To extract the time from “226218”

- Subract 11 from the first two digits (22) to get the hour (11).
- Subtract 11 from the next two digits (62) to get the minutes (51).
- Subtract 11 from the last two digits (18) to get the seconds (7).

So the build time in this example is 11:51:07.

Where else is version information available?

In the Administration interface, click the server instance name in the left panel and select the General tab. The Install Version information appears in the right panel.

At the command line, use the `asadmin version` command:

```
version [--user admin_user] [--password admin_password] [--host
localhost] [--port 4848] [--local=false] [--verbose=false]
```

You can also check the `/var/sadm/install/productregistry` XML file for the Application Server.

How do I access the sample applications?

You can tour the features of the Sun Java System Application Server by running the sample applications. Access the start page here:

install_dir`/samples/index.html`

Licensing

This section addresses the following questions:

How do I renew an expired evaluation license?

Update the license using the `asadmin install-license` command.

Instructions are contained in the Licensing appendix of the *Sun Java System Application Server Installation Guide*

Can I upgrade from an evaluation license to a full Sun Java System Application Server 7 license?

Maybe. Follow these steps:

1. Go to the Application Server's home directory.
2. Go to the `/bin` dir.
3. Run the `asadmin` command.

   This starts a new session with the prompt `asadmin>`
4. Enter the command `display-licence`.

   The licence details are displayed.
5. Check whether you have unlimited licence. If you don't, you cannot upgrade.
6. Enter the `install-licence` command.

   You will be prompted for your key.
7. Copy and paste the key from the following location:

   `http://edist.central/`

In a media package, how do I get the license key?

The software license key, also known as the activation code, is on the welcome card in the media package.

How can I get an evaluation license for Enterprise Edition?

there isn't one

Platform Support

Most questions about supported platforms for Sun Java System Application Server 7 can be found in the *Platform Summary* documents for the various releases here:

`http://docs.sun.com/db/prod/s1.asse`

This section addresses the following questions:

What's the difference between Standard Edition and Enterprise Edition?

The Enterprise Edition (EE) provides the High Availability features like load balancing (LB) and Session Persistence (SP), in addition to all SE functionality. Please check the release notes of the EE version for specific details of LB and SP.

What Sun ONE Message Queue versions are supported?

Which versions of Sun ONE Message Queue are supported with which versions of Sun Java System Application Server?

- Sun Java System Application Server 7.1 is bundled with Sun ONE Message Queue 3.5 SP1.

What J2SE versions are supported?

- Sun Java System Application Server 7.1 is bundled with J2SE 1.4.2_04.

What database drivers are supported?

The Application Server 7 product is designed to support connectivity to any database management system with a corresponding JDBC driver. Refer to the *Sun Java System Application Server Platform Summary* for a list of the components that Sun has tested and found to be acceptable for constructing J2EE-compatible database configurations.

Additional driver have been tested to meet the JDBC requirements of the J2EE 1.3 platform with the following JDBC Driver Certification Program:

`http://java.sun.com/products/jdbc/certification.html`

These drivers can be used for JDBC connectivity with the Application Server. Although Sun offers no product support for these drivers, we will support the use of these drivers with the Application Server.

Is JADO supported for Sun Java System Application Server 7?

Answer: Java Data Objects (JADO) is used under the covers for the container-managed persistence (CMP) engine. JDO is not exposed / supported.

Does Sun Java System Application Server 7 have a JCA connector to talk to SAP R3?

Our ISV partners for JCA connectors are Insevo, Attachmate, iWay, and Seagull. They have ported to Sun Java System Application Server 7.

Are custom authorization providers supported?

As a J2EE 1.3 product, Sun Java System Application Server 7 does not support custom authorization providers. Custom authorization plug-in support is defined by JSR-115 (JACC) which was only introduced in J2EE 1.4. This will be supported in a future release.

How is load balancing supported?

The web connector on a web server that fronts the Application Server cannot do load balancing at all. However, a load distribution capability is enabled. The connector can interrogate the URI in the request and determine which of several Sun Java System Application Server instances is to service the request. The mapping of the URI to an application server instance is a fixed 1:1 relationship. That is, requests sent to `http://www.abc.com./`*`xyz`*`/...` will always be distributed to the single Sun Java System Application Server instance that services the *xyx* URI.

You can have authentic load balancing by using a third-party product to front the web server or the actual application server instances themselves. If HttpSession is used in the application, you will want to turn on sticky load balancing based upon embedded tags or cookies. Before a session is established, the third-party load balancer could use techniques like round robin and weighted round robin load balancing to direct traffic to any of the Sun Java System Application Server instances. Thereafter, with sticky enabled, the load balancer will *stick* all subsequent processing to the application server instance that created the original session.

Functional Support

How do you run a CMP application with a different database?

Assume you already have a CMP `ear` file using `dbschema` in oracle. In order to run the same cmp application file with Sybase, MS SQL, or other databases, perform the following operations:

1. Create the necessary tables required by the CMP application in the new database.

2. Register a JDBC resource, connection pool, and a persistence manager for the new database.

This procedure saves time by creating the dbschema for the new database.

Can remote EJB clients use something other than RMI over IIOP, or is that the only support protocol?

According to the J2EE spec, RMI / IIOP is the proposed way of remotely interacting directly with an enterprise bean inside an EJB container. However, you can wrap that bean as a web service and access it using SOAP / HTTP (SOAP over HTTP). You can also use a MOM approach using JMS destinations (queues and topics) as another way of interacting with the functionality of the bean.

Is sticky load balancing supported?

It is supported for SE, but no failovers occur.

Yes for EE. Sun Java System Application Server 7, Enterprise Edition load balancer uses a sticky round robin algorithm to load balance incoming HTTP and HTTPS requests. The load balancer plug-in uses the following methods to determine session stickiness:

- Cookie-based method—The load balancer plug-in uses a separate cookie to record the route information. The browser must support cookies.
- Explicit URL rewriting method—Sticky information is appended to the URL. Works even if the browser does not support cookies.

Are multiple JVMs for hosting components supported?

With both PE and SE, you can create multiple application server instances for a given product installation. By default, the first server instance will be named server1. You can then use the Administration interface to create a second instance called, for example, server2. Once started, both server1 and server2 will have their own separate JVMs. However, a given instance (such as server1) has one and only one JVM. Instances do not share JVMs.

General Administration

This section addresses the following questions:

How do I logout from the Admin Console?

There is no explicit logout button in the Admin Console. To logout , close the browser you are using to access the console. Also, close other instances of the same browser that may be running on your machine. For example, if you are using Netscape 4.79 to login and access the Admin Console, you need to close all instances of Netscape 4.79 running on your machine. That action will log you out of the admin GUI.

How do I give limited admin privileges to a user?

Use the following procedure to give admin privileges to another user for a particular instance of the server.

1. Stop the appserver instance (`server1`)
2. Make sure that the `imq` broker instance is also stopped (`ps -ef | grep imq`).
3. Change Owner recursively for all the directories and subdirectories including `server1` under `/domains/domain1` (so that the new user is now the owner of those directories), using a command like this: `chown -R s1as server1`
4. Change the Group recursively for all the directories and subdirectories including `server1` under `/domains/domain1` (so that the group is now changed to the group under which the new user is a member of), using a command like this: `chgrp -R appserver server1`
5. Go to `/domains/domain1/server1/config`
6. In the `init.conf` file, change the user name from root to the new user.
7. Make sure that these changes are reflected, using `asadmin reconfig`
8. Repeat steps 4 and 5 for the iMQ instance also. This instance is found under `/var/imq/instances`. The user and group name should now be changed for all the directories and sub-directories including `domain1_server1`.

9. Start the appserver instance (`server1`) by logging in as the new user.
10. Deploy an application into this server instance.

The owner of this application is now the user.

High-Availability

This section addresses the following questions:

- Where are the load balancer files?
- How do I enable the health checker?

Where are the load balancer files?

The load balancer configuration files, `sun-loadbalancer_1_0.dtd` and `loadbalancer.xml`, should be in the following locations:

- For Sun ONE Web Server—*webserver_install_dir*`/`*webserver_instance*`/config`
- For Apache Web Server—*webserver_install_dir*`/`*webserver_instance*`/conf`

How do I enable the health checker?

To enable the health checker for the load balancer, edit the following properties int the loadbalancer.xml file:

- `url`—Specify the URL of the listener that the load balancer will check.
- `interval-in-seconds`—Specify the interval at which health checking will occur. Default is 30 seconds.
- `timeout-in-seconds`—Specify the timeout interval for a response to be received from a healthy listener. Default is 10 seconds.

Server Configuration and Control

This section addresses the following questions:

- Is instance restart the same as stop / start?

Is instance restart the same as stop/start?

No. There is a significant difference between issuing the `restart` command, and issuing consecutive commands to stop and then start an instance.

The `restart` can be used to restart an instance on the same JVM with the same security environment. So restarting an instance will not change JVM configuration parameters (like Debug=on / off) or security parameters (like security-enabled). To make such changes effective, you must stop the instance and then start it again.

The restart command is convenient, in that it allows an admin to restart a server without re-typing security passwords. So it is useful for changing the instance configuration. It also lets a server instance be started by a manager who knows the security passwords, while allowing it to be restarted by an admin.

How to supply password for automated startup or unattended restart?

By default, the Application Server prompts the administrator for the SSL key database password before starting up. To allow unattended restart, create a filed called `password.conf` in the `instance/config` directory, andinsert the following information:

```
internal:<adminPassword>
```

Unattended restart is then enabled, and the `instance/bin/startserv` script can then be used to to start the server.

NOTE Make sure that your system is adequately protected so that this file and the key databases are not compromised.

How do you specify an XA DataSource in a jdbc-connection-pool?

To use an XA DataSource, set `res-type="javax.sql.XADataSource"`. For example:

```
<jdbc-connection-pool name="myPool"
  ... res-type="javax.sql.XADataSource">
```

If this attribute is not set, the pool will default to non-xa if the class also implements DataSource. Otherwise, an exception is thrown.

How do you setup a hosts virtual server?

The Application Server has two types of virtual servers: hosts and ip virtual server. To setup a hosts virtual server, use one of the following procedures:

- Use the Admin Console:
  - **a.** Launch a browser window from the machine you want to add the hosts virtual server on.
  - **b.** Use that browser to access the Admin Console.
  - **c.** Select server1 -> HTTP Server -> Virtual Servers.
  - **d.** Click New and set the following values:

- Id: vs1
- Hosts: abc.com
- HTTP listeners: Select the default listener (`http-listener-1`) or specify a new listener

Note:
A remote browser can't be used to set up "abc.com" unless it is lauched from the remote machine -- because the admin console can only add a hosts virtual server into its own `/etc/hosts` environment.

- On aUnix platform:

  a. Login as root user.

  b. Open `/etc/hosts` and add the following line:

  192.18.12.165 abc.com

- On a Windows platform:

  a. Login as administrator.

  b. Open c:/winnt/system32/drivers/etc/hosts and add the following:

  ```
  192.18.12.165 abc.com
  ```

How do you disable batch insert and update?

The cmp solution uses batched inserts and updates to optimize performance. This is a standard JDBC feature. However, there might be some database drivers that do not implement or fully support this feature correctly.

To disable batched inserts and updates, please define following in the `java-config` section of the `server.xml` file:

```
<jvm-options>
-Dcom.sun.jdo.spi.persistence.support.sqlstore.USE_BATCH=false
</jvm-options>
```

How do you add a new user and password?

Use the admin console to add a new user and password to the key file for asecurity application. Access the admin console at `http://localhost:<admin_port>`, andperform the following operations:

1. Expand the server instance (for example, server1).
2. Expand Security.
3. Expand Realms.
4. Open File.
5. Click Manage Users.
6. Click New and enter the required information (User ID, Password, Retype Password, Group List).
7. Click OK.
8. Click the server instance again.
9. Apply changes and restart the instance.
10. Go to the `~/config/keyfile` and verify that the entry was added.

How do you set the JVM heap size?

Use the jvm-options element in the`java-config` section of the `server.xml` file as follows:

```
<jvm-options> -Xms104857600 -Xmx454857600 </jvm-options>
```

For a list of options, see `java -help` and `java -X`.

How do you start a single application server instance rather than all the instances in a domain?

To start *individual* instances in the default domain (domain1), use the `asadmin start-instance` command as follows:

```
start-instance server3
```

To start *all* instances in the default domain (domain 1), use the `asadmin start-appserver` command.

What is the ‘Apply Changes Required’ message about?

What is happening when I receive the Apply Changes Required message in the Administration interface? I made my changes and pressed the Save button. Why do I have to also Apply Changes?

When you make your changes to a configuration setting for an instance and click Save, the update is written to a temporary file called `server.xml.changes` in a backup subdirectory under the `/config` directory. The actual `server.xml` file, stored in the main `/config` directory, represents the current configuration of the server. This file has yet to be updated. When you "Apply Changes" as prompted, the contents of the `server.xml.changes` file are applied to the `server.xml` file in the main `/config` directory. When the update is successful, the `server.xml.changes` file is deleted from the `/config/backup` directory.

This is a good feature because it allows you to make a number of configuration changes without needing to apply them to the running server one at a time. It is essentially a batch update feature.

NOTE The Administration interface tries to determine what configuration changes require a server restart. If you have changed several configuration settings that require a server instance restart, Apply Changes allows you to apply them all at once, requiring only one server restart.

Why are multiple IIOP listeners useful and how many listeners are allowed?

When configuring the ORB in the Admin Console, you can have multiple IIOP Listeners. The intention is to support an administrator's desire to have an application server instance support one port for “plain” IIOP and another port for IIOP/SSL. The Administration interface lets you configure as many IIOP Listeners as you want, but only two can be enabled (Listener Enabled toggled on) at the same time. If you try to have more than two enabled at once, the Administration interface displays an error.

Does an application server instance have to be running in order to deploy to it?

No, the instance need not be running, but the Admin Server that controls the instance needs to be up and running. If the Admin Server is running, you can either use its Deploy feature in the Administration interface or the `asadmin deploy` command to deploy a module or an application to an instance regardless of whether or not the instance is up and running.

Does the Admin Server need to be running to run my application?

No. You can exercise an application server instance without the Admin Server process running. However, to utilize the SNMP capabilities for monitoring the HTTP server, you need to have the Sun Java System Application Server administration process up and running due to its role in interacting with the SNMP master agent.

TIP	After you add an HTTP listener to an application server instance, you must restart the SNMP monitoring subagent. If you do not, it's possible that information about the new listener will not be available through the SNMP monitoring subagent.

Can I configure the application server to run as non-root?

Instructions for setting up administration for a non-root user are contained in the *Sun Java System Application Server Installation Guide.*

Where is the secondary storage area for stateful session beans?

In an Entity bean the passivation process has to be considered as a simple notification that the instance is going to be dissociated from the object. Saving the bean state (synchronizing it with the database), on an entity bean, is something that is done through `ejbStore()`.

For stateful session beans, the passivation process of a bean saves the state of the bean in a secondary storage. The database is considered to be the primary storage area. The stateful container's own storage area (memory, a cache, or the disk) is the secondary storage area.

To find the secondary storage area on the Application Server from the Admin Console:

- Click the instance server on the left frame, for example the default instance, server1.
- From right frame, click Advanced.
- An element called Session Store Location contains the file system location of the secondary storage area for stateful session bean. (You can also change the directory here, but after doing so, you need to apply changes and restart the instance server.)

When you deploy a J2EE application on the instance server, a directory with the application name is created under `"Session Store Location"/ejb/j2ee-apps`. Under this application name directory, a sub directory with full qualified bean class name is created. If you deploy an ejb module, the same directory structure is created under `"Session Store Location"/ejb/j2ee-modules`.

Refer to the Application Server developer guide to find out how a stateful session bean is eligible for passivation, and refer to `sun-ejb-jar_2_0-0.dtd` to find out how to customize bean-cache properties.

When the stateful bean is passivated, stateful container will write out the serialized state of the bean into a file under secondary storage area. This file will be kept on disk until the next call comes in with same session id. The file will be removed if user calls `ejb.remove()` on this bean, or it will be removed by the stateful container when <*`removal-timeout-in-seconds`*> is reached. If you restart the instance server, all objects in the secondary storage will be cleaned up.

Application Programming

This section addresses the following questions:

How does an application implement transaction timeouts?

In the current release there is no way to set a transaction time out for container managed transactions. The default is no time out. To time out transactions, use bean managed transactions. Modify `server.xml` to change the default transaction time out value for all beans:

```
transaction-service timeout-in-seconds="0"
```

How do I use Xerces versions of JAXP in my application?

Version 7 of the application server relies on the Crimson version of the XML parser (JAXP 1.1) that is built into the JVM. The Xerces versions are included in the installation at `<install_dir>/lib/endorsed`, but they can't be used as an endorsed standard.

Typically, a JAXPapplication sets the `endorsed.dirs` system property to override the libraries found on the boot class path. But that technique does not work for a Version 7 server application, because doing so would break the container.

Tthe Xerces versions of the JAXP libraries are included in the server's classpath, however. To access them, the application must configure the JAXP factory using the following property names and associated values:

```
property: “javax.xml.parsers.DocumentBuilderFactory”,
value:    "org.apache.xerces.jaxp.JAXPDocumentBuilderFactoryImpl"

property: “javax.xml.parsers.SAXParserFactory”,
value:    "org.apache.xerces.jaxp.SAXParserFactoryImpl"
```

How do I get an instanceTransactionManager?

The `TransactionManager` is a privileged interface. Applications should only use `UserTransaction`, and not `TransactionManager`.

Application Debugging

This section addresses the following questions:

Can I modify roles in the web or EJB deployment descriptors without restarting the application server?

Changes to application deployment descriptors require a redeploy (not necessarily a restart, if you are using dynamic redeployment).

However, changing application roles dynamically is not really the best approach. The application roles in J2EE are intended as design-time groupings. Instead, look into dynamically changing the mapping of particular users to these roles. You can do this by mapping the J2EE application role to a group (or groups) and altering the membership of these groups as needed.

Upgrade/Migration

This section addresses the following questions:

Can I upgrade from other versions of Sun Java System Application Server 7 to Enterprise Edition?

Yes. You can use the automatic upgrade tool to do so.

Note:
Only one version of Sun Java System Application Server can be on a single machine, so if you have an existing version 7 installation, you must first uninstall your existing version using the uninstallation program, then install Enterprise Edition.

Can I migrate from other application servers?

Yes. You can use the automatic migration tool to do so.

Index

SYMBOLS

NUMERICS

A

B

C

D

E

F

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X